KASPERSKY lab

Den Flaschenhals schließen: So wehren Sie einen Exploit ab

www.kaspersky.de
#truecybersecurity

Den Flaschenhals schließen: So wehren Sie einen Exploit ab

Exploits als Teil der modernen Bedrohungslandschaft

Trotz einiger neuer Trends wie z. B. der verstärkte Einsatz von Komponenten, die nicht auf Malware basieren, sind cyberkriminelle Angreifer grundsätzlich weiterhin auf die gängigen Techniken zur Ausnutzung von Schwachstellen angewiesen, der primären Methode für eine erste Penetration. 2016 nahm die Nutzung von Exploits insgesamt um rund ein Viertel zu, bei Unternehmensbenutzern allein war der Anteil sogar noch höher.

Die am weitesten verbreiteten Szenarien bleiben auch weiterhin: Entweder Anhänge mit Exploits in E-Mail-Benachrichtigungen oder Drive-by-Angriffe einschließlich schädliche Links und „Watering-Hole-Umleitungen" auf dieselben Links mit einsatzbereiten Exploits, die jede Schwachstelle im Zielsystem angreifen. Dabei ist seit Langem klar, dass Angreifer zufällig einen Zero-Day-Exploit nutzen könnten, meistens aber bereits bekannte Schwachstellen ausnutzen. Tatsächlich ist die Wahrscheinlichkeit, ein nicht gepatchtes Betriebssystem oder eine noch genutzte alte App zu finden, hoch genug, um diese Angriffe lohnenswert zu machen. So ist etwa die bekannte, ehemals von Stuxnet verwendete, Schwachstelle CVE-2010-2568 weiterhin für die höchste Anzahl angegriffener Benutzer verantwortlich. Die jüngste durch die WannaCry-Ransomware verursachte Pandemie nutzte ebenfalls eine Schwachstelle für die ein Patch verfügbar war. Andererseits verwenden auch extrem sicherheitsbewusste Unternehmen anfällige Software in unternehmenskritischen Prozessen. Das kann verschiedene Gründen haben, u. a. komplexe Updates, Kompatibilitätsprobleme oder ältere Programme mit sensbiler Konfiguration.

Daher ist der Schutz vor der Ausnutzung von Schwachstellen weiterhin ein zentrales Anliegen der Endpoint-Sicherheit. Schließlich hängen davon die Präventionsmaßnahmen von Datenschutzverletzungen ab, und wie effektiv der neue Angriffe abgewehrt werden können.

Die „Kill Chain" des Exploits und Gegenmaßnahmen in jeder Phase

Für den Endbenutzer zählt natürlich nur das bestmögliche Ergebnis: Wenn die bestehende Sicherheitslösung den Angreifer erfolgreich an einer bösartigen Aktivität hindert, ist das ein Sieg.

Für den Anbieter der Lösung ist diese Phase eines Cyberangriffs jedoch ein sehr entscheidender Moment: Es geht dabei um die Programme des Benutzers, bei denen sich der unbedachte Umgang sehr schnell auswirkt, und mit Abstürzen oder dem „Blue Screen of Death" (Blauer Bildschirm) quittiert wird. Jede Phase einer Exploit-Kill-Chain bietet dem Verteidiger unterschiedliche Chancen und Herausforderungen. Sehen wir uns das im Detail an:

Zustellung

Typischerweise ein E-Mail-Anhang oder eine Webseite, wie oben beschrieben. Das führt dann dazu, dass das betroffene Programm die gebotenen Daten verarbeitet – einschließlich des Exploit-Codes.

Gegenmaßnahmen

Einige Exploits können während dieser Phase mit einem entsprechenden Mail-Server-Schutz, Anti-Phishing- und Inhaltsanalysen abgewehrt werden. Eine beträchtliche Menge an Massen-Malware wird hier tatsächlich blockiert. Hoch entwickelte Exemplare, insbesondere bei gut vorbereiteten zielgerichteten Angriffen, können statische Analysemechanismen jedoch auf eine falsche Fährte locken. Die dynamische Untersuchung aller eingehenden Inhalte in einer Sandbox ist eine gute Strategie, erfordert aber erhebliche Ressourcen und Kenntnisse, um wirklich effektiv zu sein. Außerdem würde sie in den meisten Fällen nicht *abwehren*, sondern die Sicherheitsbeauftragten des Unternehmens lediglich *alarmieren*. Angesichts der Zeitspanne von Eingang und tatsächlicher Sandbox-Detonation (aufgrund der gewöhnlichen Warteschlange) ist dies kein echter *Schutz* vor Exploits.

Speichermanipulation

Während dieser Phase werden schädliche Daten in verschiedene Speicherbereiche geschrieben. Dies ist kein Verstoß gegen Sicherheitsprinzipien und an sich meist harmlos. Zu einem späteren Zeitpunkt, wenn die Schwachstelle ausgenutzt ist, werden diese Daten jedoch in spezifischen Angriffsprozessen genutzt.

Gegenmaßnahmen

Es gibt nicht allzu viele Möglichkeiten, diese schädlichen Daten einzuschleusen, und alle sind bekannt, sodass moderne Betriebssysteme integrierte Abwehrmaßnahmen bieten, um Exploits in dieser Phase entgegenzuwirken. Damit diese Schutzvorrichtungen jedoch wirksam sind, müssen Programme mit bestimmten Parametern in einer modernen Entwicklungsumgebung kompiliert werden. Leider ist das für einige ältere Programme nicht möglich. Einige Abwehrmaßnahmen können extern ausgeführt werden, doch die Kehrseite der Medaille ist, dass eine externe Beschränkung der Speichernutzung zu Instabilitäten und Abstürzen der Programme führen kann, die von der Sicherheitslösung geschützt werden.

Ausnutzung

Hier beginnen die Aktivitäten, die nicht Teil des üblichen Ablaufs sind. Über die ausgenutzt Schwachstelle zwingt der Angreifer den Prozess des angegriffenen Programms dazu, Aktionen auszuführen, die zwar unter Umständen innerhalb dessen Standardfunktionen liegen, aber in jedem Fall für den Fortschritt des Angreifers ausschlaggebend sind. Je nach Angriffsschema folgt in der Regel eine Shellcode-Ausführung. In einigen Fällen kann jedoch die Standardfunktionalität des Programms genutzt werden, um die Malware-Nutzlast vom C&C-Server aufzurufen.

Gegenmaßnahmen

Um Aktivitäten in dieser Phase erkennen und beeinflussen zu können, muss die Sicherheitslösung auf den Kontext des geschützten Prozesses zugreifen können. Die einzige effektive Möglichkeit ist die Ausführung einer Prozessinjektion, ähnlich der Malware-Technik selbst. Mit diesem Ansatz können Exploits zwar früher gestoppt und willkürliche Prozesse geschützt werden, er hat aber auch erhebliche Nachteile. Oft kommt es zu Leistungseinbußen und Kompatibilitätsproblemen, und deren Wahrscheinlichkeit steigt mit jeder für einen bestimmten Prozess eingesetzte Abwehrmethode. Bei Prozessen, für die die Lösung zuvor nicht getestet wurde, kann die Notwendigkeit einer langwierigen Trial-and-Error-Konfiguration außerdem zu großen Unannehmlichkeiten führen, insbesondere für IT-Administratoren, die keine Spezialisten in diesem Bereich sind. Einige Anbieter empfehlen dringend, sich an deren Support-Teams zu wenden, bevor Administratoren versuchen, Sicherheitsregeln selbst anzupassen.

Hinzu kommt, dass jede neue Programmversion zu unerwarteten Abstürzen führen kann und somit die Sicherheitseinstellungen angepasst werden müssen, um eine sichere Konfiguration zu finden. Oder aber dieser Mechanismus kann nicht mehr verwendet werden, bis der Lösungsanbieter ihn ausreichend anpasst (falls dies überhaupt geschieht).

Shellcode-Ausführung

In dieser Phase wird der willkürliche Code des Angreifers und in weiterer Folge eine schädliche Payload ausgeführt.

Gegenmaßnahmen

Hier beginnt der ausgenutzte Prozess unerwartete Aktivitäten auszuführen, die er eigentlich nicht sollte. Dies kann von außen durch nicht-invasive Mechanismen zur Verhaltensverfolgung erkannt werden. Diese Mechanismen erfordern in der Regel keine manuelle Konfiguration, wodurch IT-Mitarbeiter viel Zeit und Aufwand sparen. Außerdem wird der geschützte Prozess selbst nicht angetastet, weshalb keinerlei Kompatibilitäts- oder Leistungsprobleme auftreten. Allerdings ist zu beachten, dass die Effektivität dieses Ansatzes nicht nur vom Wissen darüber abhängt, welche Aktivitäten als verdächtig eingestuft werden sollten, sondern auch vom Wissen darüber, was der Prozess *normalerweise tut*. Daher eignet er sich kaum für zuvor nicht bekannte Programme. Andererseits haben 99,9 % der Exploit-Szenarien eine recht kleine Zahl beliebter Programme und Systemkomponenten zum Ziel, sodass die Vorteile einer einfachen Verteidigung gegenüber den Einschränkungen hier deutlich überwiegen. Dieser Ansatz kann durch zusätzliche Quellen für Threat Intelligence, wie z. B. Listen mit auf gezielte Angriffe ausgerichtete Hosts und IP-Adressen außerdem noch wirksamer werden.

Payload-Ausführung

Die Payload kann als Datei heruntergeladen werden – oder bei einem vollständig körperlosen Szenario direkt in den Systemspeicher geladen und ausgeführt werden. Danach startet die schädliche Aktivität.

Gegenmaßnahmen

Das Starten eines weiteren Programms oder eines weiteren Ausführungs-Threads kann verdächtig aussehen, insbesondere wenn bekannt ist, dass das betreffende Programm diese Funktionalität nicht bietet. Dies kann also durchaus als Vorwand für eine Sicherheitslösung dienen, um die Ausführung zu „pausieren" und die Legitimität des Starts genauer zu analysieren. Zusätzliche Verhaltensindikatoren von Mechanismen zur Ausführungsüberwachung sollten der Lösung dann ermöglichen, die Ausführung der Nutzlast sicher abzuwehren.

Diese Phase gilt für alle Auswertungsszenarien – das Ziel jedes einzelnen Exploits ist letztendlich, eine Payload zu starten. Dies führt zu einem Engpass und der Angreifer hat nur sehr wenig Handlungsspielraum. Trotz der Tatsache, dass der Exploit bereits ausgeführt wurde, ist die gesamte Angriffssequenz an diesem Punkt am anfälligsten.

Wie Sie sehen, können verschiedene Phasen der Exploit-Micro-Kill-Chain unterschiedliche Gegenmaßnahmen erfordern. Während wir bei Kaspersky Lab einen mehrstufigen Cybersicherheitsansatz als den effektivsten betrachten, verstehen wir auch, dass das beste Ergebnis für unsere Kunden nicht nur zuverlässiger Schutz ist, sondern auch die Auswirkung auf bestehende Geschäftsprozesse gering gehalten werden müssen.

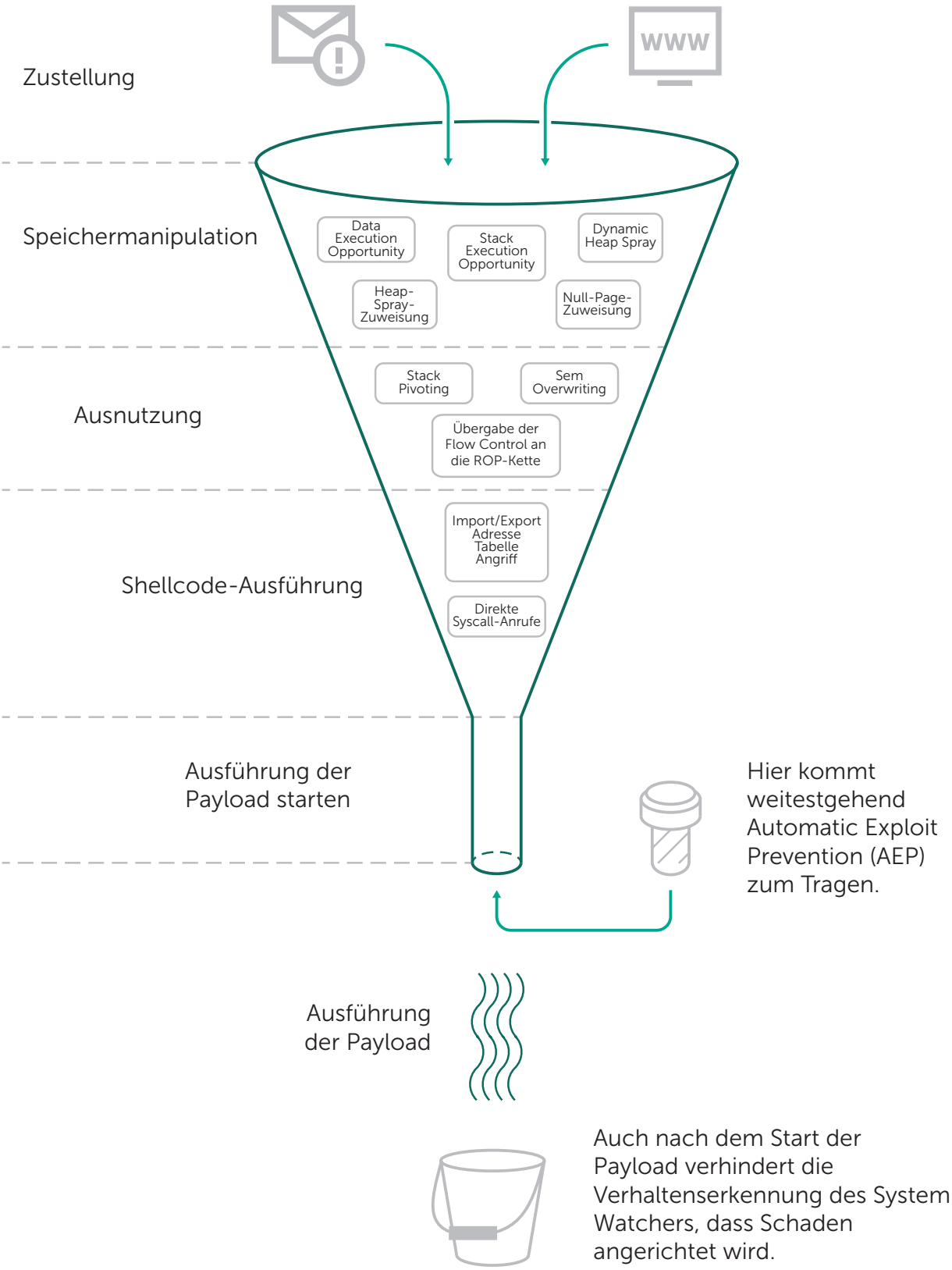
Deshalb haben wir unsere Automatic Exploit Prevention (AEP)[1] entwickelt, ein mehrschichtiges System, das nicht nur eine wirksame, sondern auch eine zuverlässige, ressourceneffiziente und einfache Kombination von Techniken verwendet, um Endbenutzern und Administratoren eine reibungslose Nutzung zu ermöglichen.[2]

Angriffstechniken, Zero-Days und Kaspersky Automatic Exploit Prevention (AEP)

Obwohl es intuitiv sicherer erscheint, einen Exploit so früh wie möglich in seiner Kill Chain zu blockieren, sind die Techniken dafür tatsächlich viel weniger wirksam, als wir uns das wünschen würden. Zahlreiche Probleme mit der Kompatibilität bzw. den Änderungen an den geschützten Programmen haben dazu geführt, dass wir uns bei Kaspersky Lab dazu entschieden haben, auf die meisten dieser Schutzmethoden zu verzichten und uns stattdessen auf nicht invasive Verhaltensprävention zu konzentrieren. Denken Sie daran, dass diese Schutzmechanismen mit bereits allgemein bekannten Techniken arbeiten. Ein Zero-Day-Exploit, der ungewöhnliche Methoden verwendet, wird diese Mechanismen also wahrscheinlich umgehen. Viele dieser Schutztechniken, die von anderen Anbietern als Kaspersky Lab verwendet werden, ähneln in der Tat denen des sehr bekannten Microsoft EMET – und es gibt mehrere Machbarkeitsstudien, die genau zeigen, wie man sie umgehen kann.

[1] Automatic Exploit Prevention ist Teil von Kaspersky Endpoint Security for Business und Kaspersky Security for Virtualization Light Agent.

[2] Sie können sich das Ergebnis der unabhängigen Tests durch MRG Effitas und AV Comparatives unter den jeweiligen Links ansehen.


Zustellung
WWW
Speichermanipulation
Data Execution Opportunity
Stack Execution Opportunity
Dynamic Heap Spray
Heap-Spray-Zuweisung
Null-Page-Zuweisung
Ausnutzung
Stack Pivoting
Sem Overwriting
Übergabe der Flow Control an die ROP-Kette
Shellcode-Ausführung
Import/Export Adresse Tabelle Angriff
Direkte Syscall-Anrufe
Ausführung der Payload starten
Hier kommt weitestgehend Automatic Exploit Prevention (AEP) zum Tragen.
Ausführung der Payload
Auch nach dem Start der Payload verhindert die Verhaltenserkennung des System Watchers, dass Schaden angerichtet wird.

Die Verhaltenserkennung andererseits nutzt eine Reihe indirekter Indikatoren. Kaspersky Automatic Exploit Prevention kann beispielsweise auch zusätzliche Informationsquellen nutzen, die von verschiedenen Sicherheitsebenen bereitgestellt werden, etwa Verfolgung von Änderungen in bestimmten Speicherbereichen, Aufruf verdächtiger URLs usw. Sowohl unabhängige Tests als auch mehrere reale Fälle beweisen, dass Kaspersky Automatic Exploit Prevention sowohl synthetische als auch echte Zero-Day-Angriffe erfolgreich erkennt, auch wenn dieser Angriff zuvor nicht bekannt war. Dank der Automatic Exploit Prevention beweisen Produkte von Kaspersky außerdem hervorragende Wirksamkeit gegen Exploit-basierte Ransomware wie CryptXXX während der frühen Aktivitätsphasen, in denen noch keine Informationen über diese Angriffe vorhanden sind.

Einige Schutzmaßnahmen erfordern allerdings keinen großen Einsatz an Ressourcen oder Prozessreparaturen und können mit bestimmten Programmen sicher verwendet werden – daher nutzt AEP sie auch.
So wehrt Kaspersky Endpoint Security mit AEP verschiedene Angriffstechniken ab (beachten Sie, dass nicht alle davon direkt mit der Ausnutzung von Schwachstellen zusammenhängen):

Schutzmaßnahme | Angriffstechnik	Kaspersky Endpoint Security – Schutz	Kaspersky Endpoint Security – Prävention	Ergebnis
DEP (Data Execution Prevention) | Buffer Overflow	DURCH BS BEREITGESTELLT (bei den meisten modernen Programmen ist DEP standardmäßig aktiviert. Das Aktivieren von DEP für Programme, für die es nicht ausgelegt ist, kann deren Funktionsfähigkeit beeinträchtigen)		Geschützt
ASLR (Address Space Layout Randomization) | Nutzung von Daten an vorhersehbaren Speicheradressen	JA		Geschützt
Stack Pivot | Wehrt Missbrauch des Stack Pointers ab	JA		Geschützt
Null-Page-Zuweisung | Null-Page-Exploit	DURCH BS BEREITGESTELLT (Externer Schutz unsicher)		Geschützt
Heap-Spray-Zuweisung | Ablage von Shellcode-Kopien an möglichst vielen Stellen im Speicher		JA	Geschützt
Dynamic Heap Spray | Wehrt Angriffe ab, die verdächtige Heap-Spray-Sequenzen ausführen		JA	Geschützt
SEHOP (Structured Exception Handler Overwrite Protection) | Überschreiben des Structured Exception Handler (SEH)	DURCH BS BEREITGESTELLT (SEHOP ist seit Windows Vista SP1 Teil des Windows-Betriebssystems (alle modernen Büroanwendungen und Browser werden standardmäßig von SEHOP geschützt)		Geschützt
ROP | Wehrt Return-Oriented-Programming-Angriffe ab		JA	Geschützt
Load Library | Verhindert das Laden von Bibliotheken aus UNC-Pfaden	DURCH BS BEREITGESTELLT (aufgrund von Kompatibilitätsproblemen werden Schutzmaßnahmen vom BS bereitgestellt – https://support.microsoft.com/de-de/kb/2264107)		Geschützt
DLL-Übernahme		JA	Geschützt
Reflektive DLL-Injektion	JA (vom System Watcher erkannt)		Geschützt

Schutzmaßnahme | Angriffstechnik	Kaspersky Endpoint Security – Schutz	Kaspersky Endpoint Security – Prävention	Ergebnis
Netzwerk-DLL | Laden bösartiger Bibliotheken durch Platzierung auf Netzwerkpfaden		JA (Missbrauch legitimer Funktionalität, wird durch die Verhaltenserkennung vom System Watcher erkannt)	Geschützt
Hollow Process		JA	Geschützt
Syscall		JA	Geschützt
VBScript God Mode			Geschützt
WoW64		JA	Geschützt
Dateilose Erkennung			
Schädliche PowerShell-Skripte		JA	Geschützt
Schädliche TaskScheduler-Aufgaben		JA	Geschützt
WMI-Abonnements		JA	Geschützt
Schädliche Skript-Ausführung über legitime ausführbare Dateien wie mshta.exe und rundll32.exe		JA	Geschützt
CVE-2013-5331 und CVE-20144113 über Metasploit		JA	Geschützt
Umgehung von Squiblydoo AppLocker	JA		Geschützt
Java-Sperre	JA		Geschützt
Programmsperre	JA		Geschützt

An dieser Stelle sollte erwähnt werden, dass besonderes Augenmerk auf der Ausführung schädlicher Skripte liegt, wie z. B. Powershell, HTA, JS/VBS, die sich als eine der beliebtesten und gefährlichsten Techniken bei der Ausnutzung von Schwachstellen erwiesen hat.

Natürlich ergeben sich einige Probleme durch rein vom Betriebssystem bereitgestellte Schutzmaßnahmen. Was geschieht in Fällen, in denen neuere Betriebssysteme oder Programme, die deren integrierte Schutzfunktionen unterstützen, nicht verwendet werden können? Viele andere Anbieter von Sicherheitslösungen weisen darauf hin, dass sie auch in diesen schwierigen Fällen Schutz bieten – wie sieht das bei Kaspersky aus?

Unser Standpunkt ist einfach erklärt: Wir haben den Einsatz dieser Schutzmaßnahmen in Erwägung gezogen und sind zu dem Schluss gekommen, dass sich das in den meisten Fällen nicht bewährt. Das gilt insbesondere für ältere Systeme, bei denen zu viele Abfragen im Benutzermodus wertvolle Systemressourcen verbrauchen und die Geschwindigkeit des Systems über das erträgliche Maß hinaus verlangsamen können. Das lässt aber noch die oben erläuterte Tatsache außer Acht, dass die meisten davon umgangen werden können.

Selbst wenn eine Schwachstelle in einer „altehrwürdigen" Version von MS Word ausgenutzt wird, wissen wir, dass wir sie sofort danach abfangen können, wenn sie sich verdächtig verhält. Das Wichtigste beim Exploit-Schutz ist schließlich, dass der Exploit die schädliche Payload nicht startet – was ja genau die Spezialität von AEP ist. In den kompliziertesten Fällen, wie etwa den jüngsten Kernel-Exploits der Ransomware WannaCry, die auf TCP-Paketen basiert, überlässt die AEP (basierend auf der Verhaltenserkennung) die Abwehr der zweiten Verteidigungslinie, dem Schutzmechanismus nach der Ausführung. Der Flaschenhals wird in jedem Fall verschlossen.

Bei diesem mehrschichtigen Ansatz ist es wenig überraschend, dass Kunden von Kaspersky Lab – solange die entsprechenden Sicherheitsfunktionen aktiviert waren – keinerlei Schäden durch die gefürchtete WannaCry-Pandemie erlitten.

Kaspersky Lab
Cybersicherheit für Unternehmen:
www.kaspersky.de/enterprise
Neues über Cyberbedrohungen: **www.viruslist.de**
IT-Sicherheitsnachrichten: **business.kaspersky.de/**

#truecybersecurity
#HuMachine

www.kaspersky.de