

Erledigt

## El Capitan und die System Integrity Protection - Was ist das und wie kann ich es ändern?

Beitrag von „griven“ vom 30. August 2017, 00:29

Normalerweise kann man die [SIP](#) aktiv lassen wenn ein Loader genutzt wird der die Extensions via Injektion in den Prelinked Kernel einschleust (Clover oder OZ machen das so). Die Magie passiert hierbei im Speicher sprich der Prelinked Kernel wird in den Speicher des Systems entpackt und dann werden die Extensions die zusätzlich geladen werden sollen einfach mit in diesen Speicherbereich gepackt (kurz in den Speicher injiziert). MacOS geht davon aus das alles was sich im Prelinked Kernel befindet auch da sein darf denn immerhin hat es zuvor den Weg dorthin gefunden sprich es findet an der Stelle keinerlei Check statt ob die Extensions signiert sind oder nicht da diese unter normalen Umständen nie den Weg dahin gefunden hätten. Von der Warte aus ist es also komplett unkritisch ob die [SIP](#) aktiv ist oder nicht....

Probleme gibt es auf PC's aber an anderer Stelle denn macOS mag den Prelinked Kernel und alles was dazu gehört gerne in einen Speicherbereich packen der zum einen unterhalb der 4GB Grenze liegt und zum anderen auch zusammenhängend sein muss und hier krankt es dann oft denn das UEFI eines PC's geht recht verschwenderisch mit den gegebenen Ressourcen um und schmeißt einfach alles mal dahin wo eben gerade Platz ist. Stark vereinfacht gesprochen wird beim Start des Rechners der verfügbare Speicher in verschiedene Bereiche aufgeteilt und diese Bereich mit Adressen versehen (Einige Bereiche dienen als allgemein verfügbarer RAM andere wiederum als geschützte Bereich in die das UEFI Treiber lädt für die installierten Geräte oder eben auch den Kernel des jeweils benutzten Betriebssystems). Über all das führt das UEFI Buch und vermerkt akribisch genau an welchen Adressen was geladen wurde es entsteht somit eine MemoryMap die Auskunft darüber gibt was an welcher Stelle im Speicher zu finden ist. Die Map kann man sich als eine Art Landkarte vorstellen auf der einige Bereiche bereits erforscht sind (also belegt) und andere frei bleiben. Anders als Windows oder Linux mag macOS seinen Kernel und alles was dazu gehört gerne in einen Bereich schieben der en Block ausschließlich dafür reserviert ist. Auf Apple Hardware ist das kein Problem da das (U)EFI eines Macs darauf optimiert ist auf PC's jedoch wird das schnell zu einem gewaltigen Problem denn deren UEFI nimmt natürlich keine Rücksicht auf solche Befindlichkeiten.

Je nach eingesetztem Bootloader gibt es mehr oder weniger effektive Fixes für dieses Problem. In der Clover Welt sind der OSXAptioFix oder OSXAptioFix2 weit verbreitete Gegenmittel beiden gemein ist das sie die MemoryMap neu organisieren oder lax gesprochen beide Fixes spielen Speicher Tetris bis hoffentlich genug freigeräumt ist um den Bedürfnissen von macOS gerecht zu werden. Das Ganze klappt meistens aber leider bei weitem nicht immer. Nachdem

Apple die [SIP](#) eingeführt hat hat sich der Speicherhunger noch mal erhöht und nur selten gelingt es den genannten Fixes genug Speicher so umzuschichten das genug Platz ist um den macOS Kernel mitsamt seiner Extensions und dem zusätzlichen Overhead der durch die [SIP](#) erzeugt wird zu expandieren. Ein probates Mittel dagegen ist es die [SIP](#) möglichst weitestgehend zu deaktivieren denn auf die Weise wird Speicher gespart und das wirklich notwendige kann geladen werden...