

Erledigt

# El Capitan kommt! Auf unsere Rechner!

Beitrag von „MacGrummel“ vom 30. September 2015, 12:37

Wer etwas intensiver liest: Griven hatte [diese Tabelle](#) auch schon veröffentlicht. Ich gebe allerdings zu, dass unsere Server-Software bei der Übersichtlichkeit hier an ihre darstellerischen Grenzen gerät..

und kennt neben an und aus auch noch eine Menge individuelle Konfigurationen die es uns ermöglicht den Schutzgrad unseren Bedürfnissen entsprechend einzustellen. Aber wie funktioniert das Ganze nun? Beim Start des Systems wertet der OS-X eigene Bootloader (boot.efi) bzw. der Kernel den Inhalt einer bestimmten NVRAM Variable aus (CSRActiveConfig) und stellt das Schutzlevel entsprechend ihres Inhalts ein. Ist die Variable nicht vorhanden oder enthält den Wert 0 bedeutet dies das die System Integrity Protection komplett aktiv ist, weicht der Wert von der 0 ab wird der Schutzgrad entsprechend der Konfiguration eingestellt. Folgende Übersicht veranschaulicht die möglichen Konfigurationen:

	hex	n/a	nvrnm	dtrace	intern	debug	pid	fs	kexts
1									
2 csrutil enabled --no-internal	00	0	0	0	0	0	0	0	0
3 csrutil enabled	10	0	0	0	1	0	0	0	0
4 csrutil enable --without kext	11	0	0	0	1	0	0	0	1
5 csrutil enable --without fs	12	0	0	0	1	0	0	1	0
6 csrutil enable --without debug	14	0	0	0	1	0	1	0	0
7 csrutil enable --without dtrace	30	0	0	1	1	0	0	0	0
8 csrutil enable --without nvrnm	50	0	1	0	1	0	0	0	0
9 csrutil disabled	77	0	1	1	1	0	1	1	1
10									
11 Other settings									
12 disabled (no internal)	67	0	1	1	0	0	1	1	1

Hier wird deutlich, das sich das Schutzlevel sehr fein einstellen lässt. Für einen typischen Hackintosh Fall (Post install) würde es zum Beispiel reichen vorübergehend den Filesystem Schutz auszuschalten und das laden unsignedlter Extensions zu erlauben. Folgt man dem oberen Beispiel ergibt sich hieraus also:

	hex	n/a	nvrnm	dtrace	intern	debug	pid	fs	kexts
1									
2 No FileSystem, Kext	03	0	0	0	0	0	0	1	1

Also ein binärwert von 00000011 oder eben ein hex Wert von 03 setzen lässt sich der Wert nun abhängig vom Bootloader auf unterschiedliche Art und Weise. Bei Clover erreicht man das indem man den Wert in den Bereich RT-Variablen unter den Punkt CsrActiveConfig in die config.plist einträgt was dann in etwa so aussieht:

```
<key>RtVariables</key>
<dict>
  <key>CsrActiveConfig</key>
  <string>0x03</string>
</dict>
```

unter Ozmosis gelingt es indem man den Wert mit dem folgenden Kommando in den NVRAM schreibt

```
nvrnm 7C436110-AB2A-4BBB-A880-FE41995C9P82:csr-active-config=03
```