Testergebnisse – 16.12.2011

HTEditor

URL: http://hte.sourceforge.net

Ein editor/viewer/analyzer - in dem sich angeblich ein Command-line-argument-bug befindet.

Konnte aber weder den original Exploit nachvollziehen und auch der Fuzzer mit ansteigender Eingangsdatenlänge und unterschiedlichen Eingangsdatenmustern kam auf keinem der Testrechner zu einem Ergebnis.

Gefuzzter Parameter: argv[1]

Durchläufe: 1000

Datenlänge: im Bereich von 3000 - 1000000

CoreHttp

URL: http://corehttp.sourceforge.net/

Ein leichter Webserver (unter anderem für embedded Systeme)

Gefuzzter Parameter: Eingangsrequestpuffer, wird direkt aus dem HTTP-Request übernommen

Durchläufe: 21 (Abbruch wegen Eingangsdatenlängen Beschränkung)

Datenlänge: im Bereich von 1 - 2000 Byte, ansteigender Länge

Durchlauf	**Länge**	**Ergebnis**
1 - 6	0 - 500Byte	Kein abnormales Programmverhalten
7 – 8	600 - 700B	Gespeicherte Werte von ebp(ebp) und eip(rip) werden überschrieben

9 – 21	800 - 2000	Segmentation Fault, Register ebp, eip und rip werden überschrieben

Anhand dieser Daten zeigt sich dass einerseits Gespeicherte Register überschrieben werden können und somit ein klassischer Stack Overflow möglich ist und andererseits eventuell auch andere Pointer Werte überschrieben werden können die dereferenziert werden.

3Proxy

URL: http://www.3proxy.ru/

Ein Proxyserer der auch für embedded Geräte geeignet ist.

Um prinzipiell die Existenz eines Fehlers nachzuweisen und dann genauere Informationen über diesen zu bekommen wurde der Fuzzing-Prozess in 2-Unterschritte aufgeteilt: Äußeres und Inneres Fuzz Target.

Äußeres Fuzz Target

Gefuzzter Parameter: Requestpuffer, direkt in der Empfangsmethode

Durchläufe: 60

Datenlänge: im Bereich von 0 - 6000 Byte, ansteigender Länge

Durchlauf	Länge	Ergebnis
1 – 21	0 - 2000B	Kein abnormales Programmverhalten
21 - 56	2000 - 5600B	Programm Terminiert mit segmentation Fault
57 - 60	5700 - 6000B	Segmentation Fault, Register rbp, rbx, r12 und rip werden überschrieben

Die Analyse zeigt dass ab Durchlauf 21 in einer aufgerufenen Methode der Stackframe bzw. Return adressen überschrieben werden. Ab Durchlauf 57 geht der Überlauf über den inneren (aufgerufenen) Stackframe hinaus und überschreibt auch den äußeren Stackframe.

Anhand der in der LogDatei aufgezeichneten Adresse kann die aufgerufene Methode (log methode) ermittelt werden.

Inneres Fuzz Target

Gefuzzter Parameter: Requestpuffer

Durchläufe: 77

Datenlänge: im Bereich von 0 - 7500 Byte, ansteigender Länge

Durchlauf	Länge	Ergebnis
1 – 21	0 - 2000B	Kein abnormales Programmverhalten
22 - 77	2000 - 7700B	Die im Stackframe gespeicherten Register rbp und rip. Werden überschrieben.

Die Analyse zeigt dass ab Durchlauf 22 register und return adresse überschrieben erden. Somit ist eine klassische Buffer-Overflow Attacke möglich