השורד שלנו דומה לשורד בראיון לשורד JMPscare אך השינויים שביצנו הם כאלובמקום לכתוב במקום בזיכרון שהוא חשוף ופגיע שינינו את מיקום הכתיבה והחבאנו אותו בצורה שיותר קשה למציאה בקריאת הקוד. בנוסף הוספנו טירגות של שורדים אחריםבכדי לצמצם את התחרות.

בנוסף רצינו לשנות את כיוון האלכסונים הנוצרים אך כששינינו את הopcode מ 29 ל 01 (מ sub לפקודה הנגדית add) קיבלנו כתתיבה בקו ישר דבר שפגע מאוד בשורד.

דבר נוסף שהוספנו הוא ההפצצות הכבדות.

MOV WORD [0x9385], 0x1210 //set value in address 0x9385

MOV ES, [0x9385] // take value from address 0x9385 into E

MOV AX, 0xcccc // set ax value to 0xcccc

mov di, 0x4000 //set value of di for attack

int 0x86 // attack 255 byte from di with ax and increase di

int 0x86 // continue attacking

xor di,di // set value 0 to di

mov ax , 0xcccc // set value of attack to ax

mov cx , 0x20 // set loop iteration to 20

start: //set label

MOV WORD [0xECEF], ax // attack JMPscare

MOV WORD [0xfdEF], ax // attack JMPscare

MOV WORD [0x40b1], ax // attack GreenSlate

MOV WORD [0x9662], ax

MOV WORD [0x9660], ax

MOV WORD [0x2162], ax

MOV WORD [0xbf53], ax // attack Asmodeus

MOV WORD [0x3273], ax // attack Asmodeus

MOV WORD [0x3717], ax

MOV WORD [0xfea3], ax // attack COVID8086

MOV WORD [0x0250], ax

MOV WORD [0x8550], ax

MOV WORD [0x2160], ax

MOV WORD [0x5a91], ax

MOV WORD [0x0091],ax

dec cx // minus from cx-1 for the loop

jnz start// jump not zero to start loop

MOV CX, 0x704

MOV WORD [0xECEF], 0x1CFF

MOV WORD [0xECED], 0xCC29

PUSH SS

POP DS

PUSH CS

POP SS

MOV SP, 0xDDF

MOV WORD [SI], 0x1CFD

MOV WORD [SI + 0x2], 0x1CFF

CALL FAR [SI]

MOV WORD [0x9385], 0x1210 //set value in address 0x9385

MOV ES, [0x9385] // take value from address 0x9385 into E

MOV AX, 0xcccc // set ax value to 0xcccc

mov di, 0x6000 //set value of di for attack

int 0x86 // attack 255 byte from di with ax and increase di

int 0x86 // continue attacking

xor di,di // set value 0 to di

MOV CX, 0x704 // set value of 0x0704 to cx

MOV WORD [0x1242], 0x1CFF // set command (call far) in address 0x1242

MOV WORD [0x1240], 0xCC29 // set command (sub sp, cx) in address 0x1240 the attack function

PUSH SS // push into stack ss

POP DS // pop from stack into ds

PUSH CS // push into stack cs

POP SS // pop from stack into ss

MOV SP, 0x0 // set value of sp to 0 for attack

MOV WORD [DI], 0x1020 // change the value of address [di] to 0x1020 (need for change di place)

MOV WORD [DI + 0x2], 0x1022 // change the value of address [di] to 0x1020 (need for change di place)

CALL FAR [DI] // change di to address 0x1240 to run from there the rest of the code.