Intel CET

And how to stop being scared of French baking

About Me

- Software Engineer at Crowdstrike
- Previously security researcher at SentinelOne
- Instructor of Windows Internals classes
- Former pastry chef
- Circus artist
- Ninja-in-training

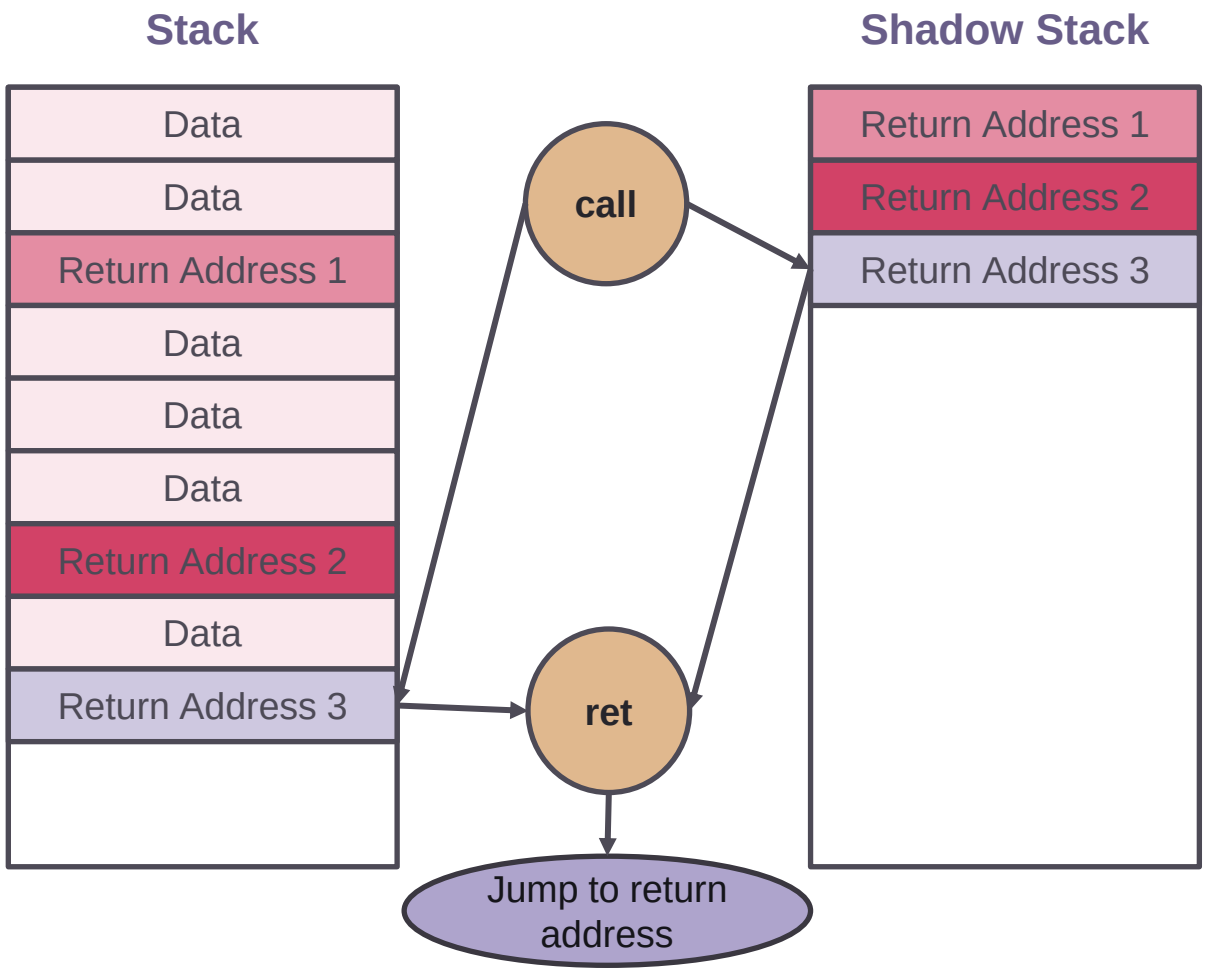
Why Intel CET?

- Common exploit technique is ROP - using stack corruption to control return addresses and gain code execution
- Another technique is overwriting function pointers to control the target or a jump or call
- Forward-edge exploitation is handled by CFG and XFG
- But ROP can’t have a software-based mitigation

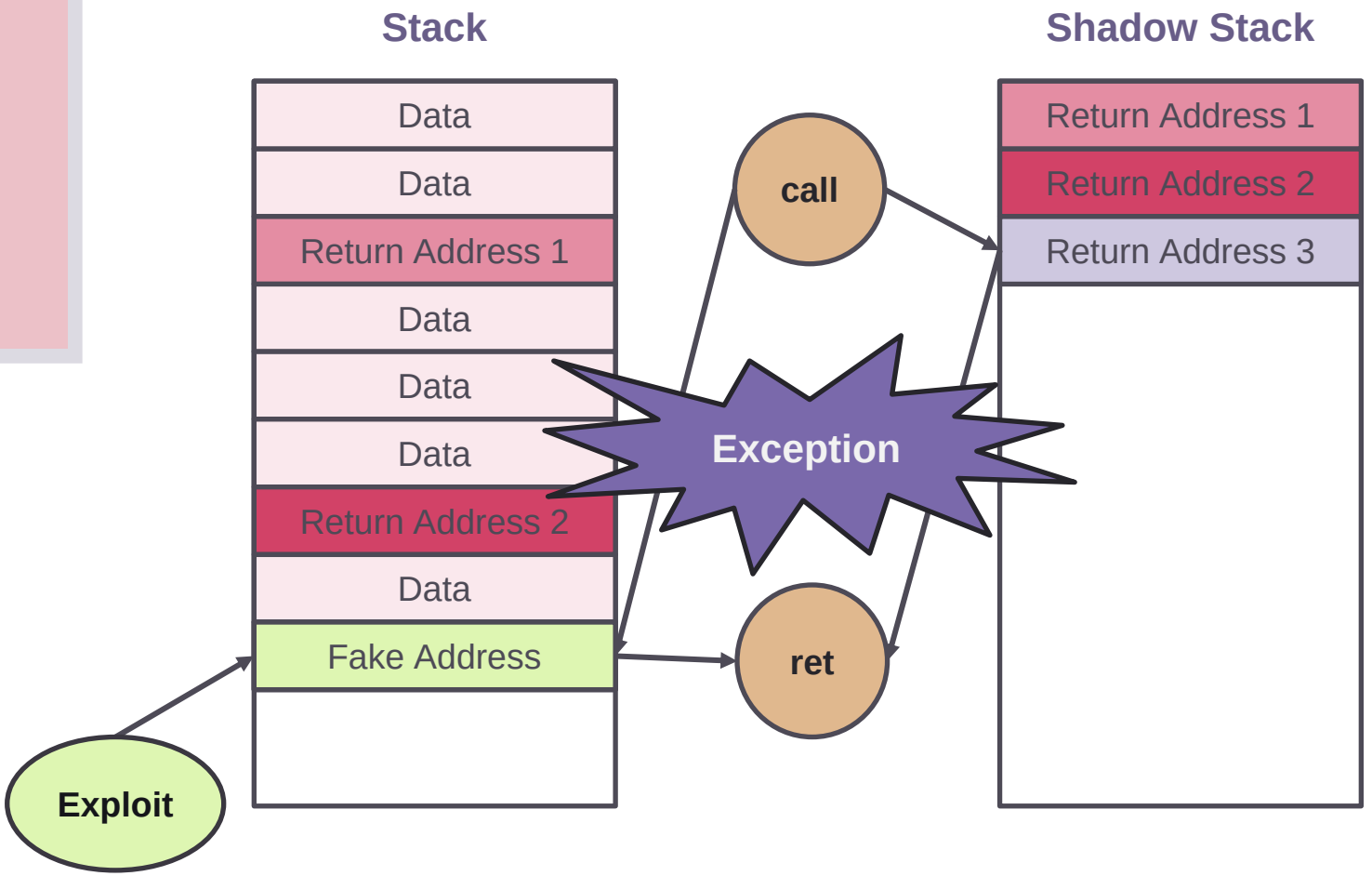
CET to the Rescue

- CET adds a second stack that only contains return addresses – called a shadow stack
- When a function call happens, return address gets pushed to both stacks
- When executing a ret instruction, processor compares return address on both stacks

CET to the Rescue



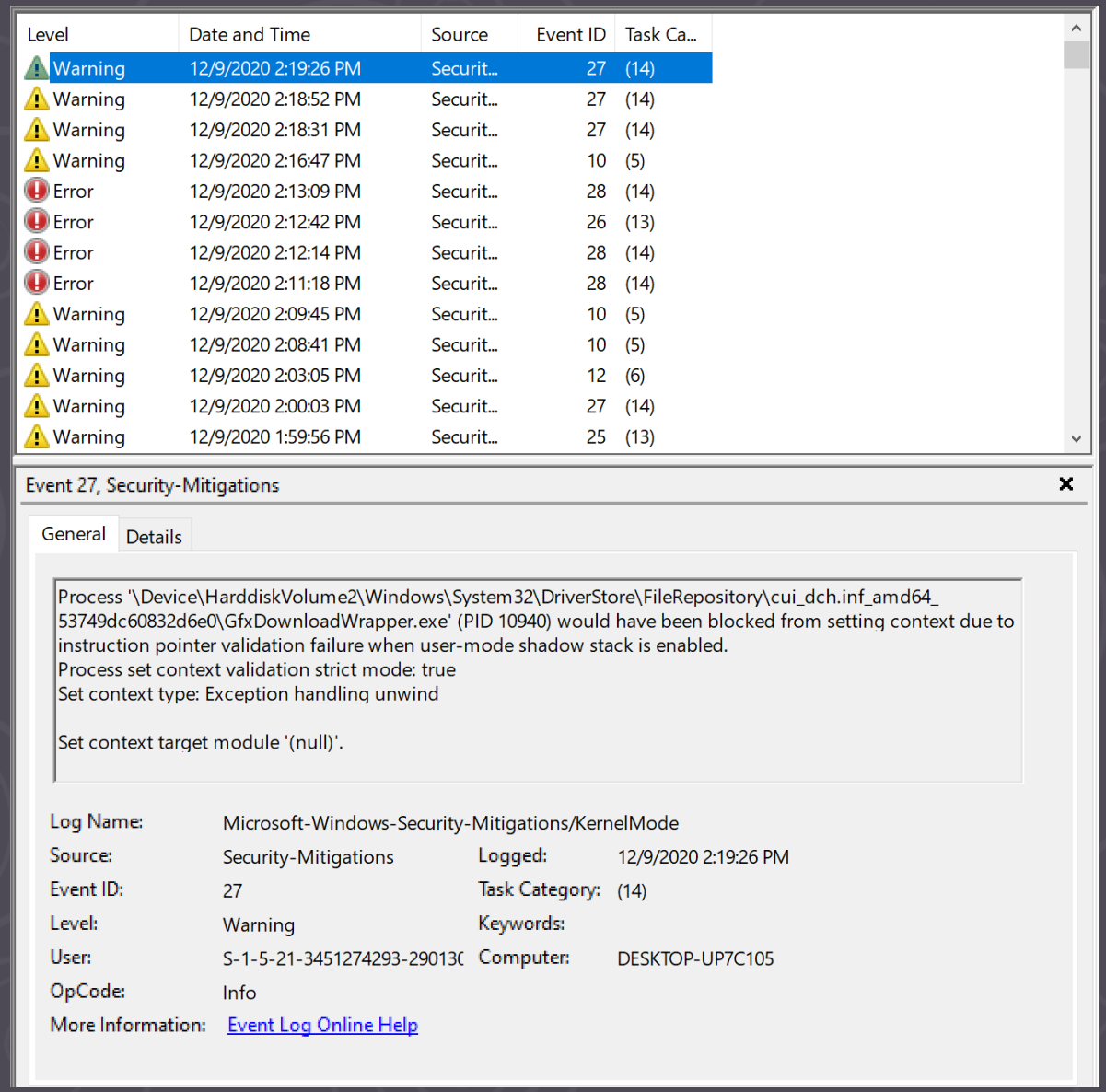
CET to the Rescue



CET Exception Handling

- The OS is in charge of catching and handling the exception
- Can choose to crash the process, log the violation, or ignore completely
- Windows will only crash processes if they were compiled with /CETCOMPAT
- Otherwise – log the error to the event log

CET Exception Handling - Audit



Level	Date and Time	Source	Event ID	Task Ca...
Warning	12/9/2020 2:19:26 PM	Securit...	27	(14)
Warning	12/9/2020 2:18:52 PM	Securit...	27	(14)
Warning	12/9/2020 2:18:31 PM	Securit...	27	(14)
Warning	12/9/2020 2:16:47 PM	Securit...	10	(5)
Error	12/9/2020 2:13:09 PM	Securit...	28	(14)
Error	12/9/2020 2:12:42 PM	Securit...	26	(13)
Error	12/9/2020 2:12:14 PM	Securit...	28	(14)
Error	12/9/2020 2:11:18 PM	Securit...	28	(14)
Warning	12/9/2020 2:09:45 PM	Securit...	10	(5)
Warning	12/9/2020 2:08:41 PM	Securit...	10	(5)
Warning	12/9/2020 2:03:05 PM	Securit...	12	(6)
Warning	12/9/2020 2:00:03 PM	Securit...	27	(14)
Warning	12/9/2020 1:59:56 PM	Securit...	25	(13)

Event 27, Security-Mitigations

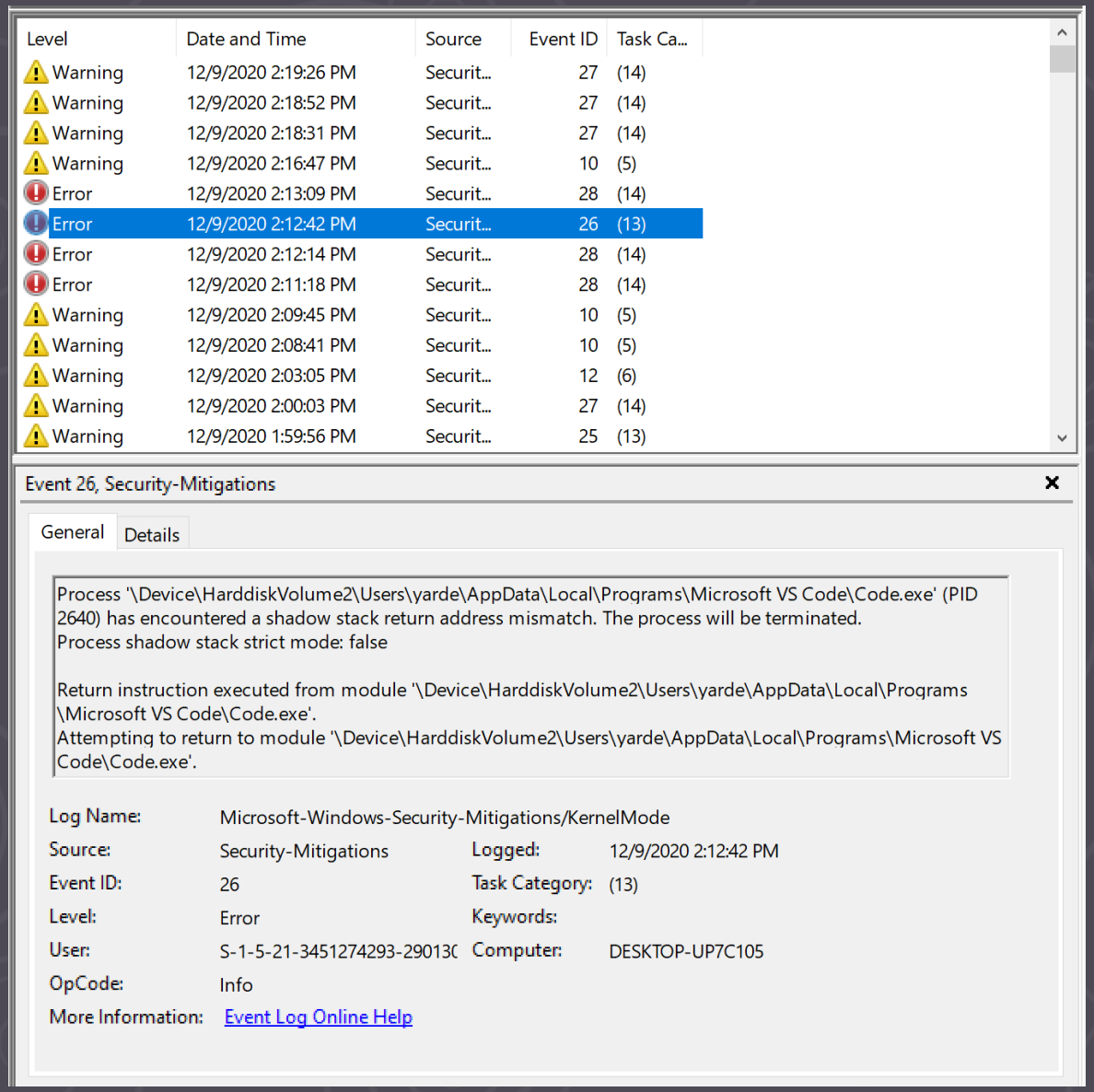
General Details

Process '\Device\HarddiskVolume2\Windows\System32\DriverStore\FileRepository\cui_dch.inf_amd64_53749dc60832d6e0\GfxDownloadWrapper.exe' (PID 10940) would have been blocked from setting context due to instruction pointer validation failure when user-mode shadow stack is enabled.
Process set context validation strict mode: true
Set context type: Exception handling unwind

Set context target module '(null)'.

Log Name: Microsoft-Windows-Security-Mitigations/KernelMode
Source: Security-Mitigations
Logged: 12/9/2020 2:19:26 PM
Event ID: 27
Task Category: (14)
Level: Warning
Keywords:
User: S-1-5-21-3451274293-29013(
Computer: DESKTOP-UP7C105
OpCode: Info
More Information: Event Log Online Help

CET Exception Handling - Crash



Level	Date and Time	Source	Event ID	Task Ca...
Warning	12/9/2020 2:19:26 PM	Securit...	27	(14)
Warning	12/9/2020 2:18:52 PM	Securit...	27	(14)
Warning	12/9/2020 2:18:31 PM	Securit...	27	(14)
Warning	12/9/2020 2:16:47 PM	Securit...	10	(5)
Error	12/9/2020 2:13:09 PM	Securit...	28	(14)
Error	12/9/2020 2:12:42 PM	Securit...	26	(13)
Error	12/9/2020 2:12:14 PM	Securit...	28	(14)
Error	12/9/2020 2:11:18 PM	Securit...	28	(14)
Warning	12/9/2020 2:09:45 PM	Securit...	10	(5)
Warning	12/9/2020 2:08:41 PM	Securit...	10	(5)
Warning	12/9/2020 2:03:05 PM	Securit...	12	(6)
Warning	12/9/2020 2:00:03 PM	Securit...	27	(14)
Warning	12/9/2020 1:59:56 PM	Securit...	25	(13)

Event 26, Security-Mitigations

General Details

Process '\Device\HarddiskVolume2\Users\yarde\AppData\Local\Programs\Microsoft VS Code\Code.exe' (PID 2640) has encountered a shadow stack return address mismatch. The process will be terminated.
Process shadow stack strict mode: false

Return instruction executed from module '\Device\HarddiskVolume2\Users\yarde\AppData\Local\Programs\Microsoft VS Code\Code.exe'.
Attempting to return to module '\Device\HarddiskVolume2\Users\yarde\AppData\Local\Programs\Microsoft VS Code\Code.exe'.

Log Name:	Microsoft-Windows-Security-Mitigations/KernelMode		
Source:	Security-Mitigations	Logged:	12/9/2020 2:12:42 PM
Event ID:	26	Task Category:	(13)
Level:	Error	Keywords:	
User:	S-1-5-21-3451274293-29013(	Computer:	DESKTOP-UP7C105
OpCode:	Info		
More Information:	Event Log Online Help		

The Art of Over-analyzing Cookies

Butter Cookies

- Made up of butter, flour and sugar
- And usually a bit of milk or egg yolk
- Delicate, crumble easily
- Can be made in any flavor and shape
- Delicious

Rule #1 – Ingredients

- Butter : Flour ration = 1 : 2

  Can be higher – I like using 2 : 3

- Butter = 82% fat, 18% liquid

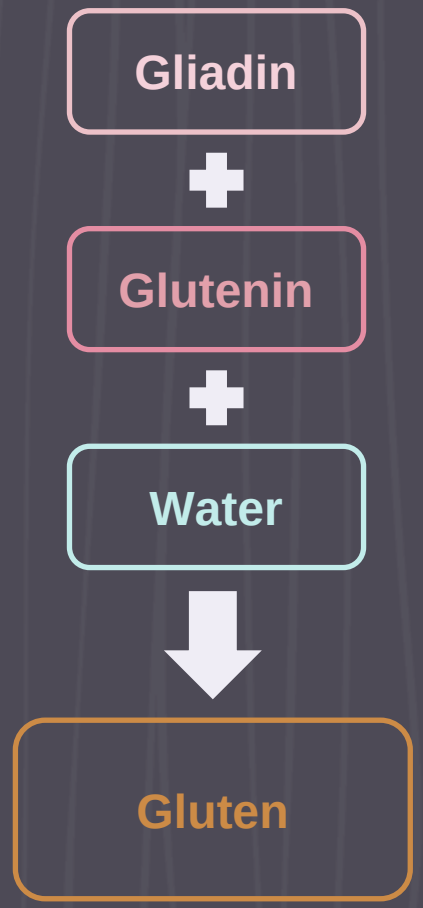
  Can be replaced by coconut oil and water – but not liquid oil

- Flour = white wheat flour

  Can be any other flour

- Add a bit of egg yolk or liquid to get everything to stick together better
- All other ingredients (including sugar!) are optional and only used for flavor

Rule #2 – Processing



- Gluten makes the dough elastic
- Makes the final (baked) product chewy
- Great for bread, not so great for cookies
- To stop gluten from being formed:
  - As little processing as possible
  - Fat – wraps around protein molecules
- Process flour, butter, sugar together for minimal time – once it starts forming dough add a bit of fat/liquid, process for a few more seconds
- Wrap dough, chill/freeze until baking

What About the Flavor?

- Sugar is only there for flavor – add as much or as little as you like
- Can replace it with dark brown sugar, maple, sugar substitute…
- Add any flavor you like – just notice the contents!
    - Cocoa powder – dry – replace 10-15% of flour
    - Coffee / orange juice / strawberry / Irish cream – liquid – add at the end instead of milk/yolk
    - Nut butter / Tahini – fat – replace 5% of butter
    - Vanilla / zests / flavor extracts / spices – very small amounts, just add it
    - Nuts / chocolate chips – add after dough starts to form
    - Or remove the sugar and use pesto and cheese for savory cookies!

Dietary Restrictions

- Gluten allergy – use gluten-free flour
- No dairy – Replace butter with 85% coconut oil and 15% liquid
- Diabetes – replace sugar with substitute