All.Net Analyst Report and Newsletter

<u>Welcome to our Analyst Report and Newsletter</u>

How to defeat LLM training using your material

A fundamental challenge in the age of LLM-based AI for those who do not want their intellectual property or confidential information learned or exploited by the technology companies that hoover everything up and use it without paying for how it came to be is to make things understandable to your actual audience and not to the AI systems. For now, there remain various approaches that work and will likely work for some time.

Encryption

This one uses keys held only by the participants, has long been studied, is widely used (or rather misused), and suffers the major problem that once you see it the computer can see it as well, and that means the ‘agentic’ AI mechanism in control of your system will be able to read and use it. Endpoint security becomes a key issue, and we don’t really have that when the major providers of hardware and software are embedding remote access, control, and other functions into their distributions. It also has all of the challenges of key distribution and storage that cryptographic systems have long had, will likely be defeated because of software and covert channel flaws in the implementations, and if quantum computing is successful enough will be defeated by that result as well. For broadcast channels this is also problematic because of the widespread availability of keys, or for systems where you have to ‘sign in’ to get the keys, the authentication and identification process is subject to attack by the AI mechanisms as well; it just adds attack surface and makes it harder to use.

Steganography

For now, steganography seems to work well. That’s because the AI is currently unable or unwilling to figure out what is readably by people vs not and is too dumb to make sense of what’s going on. Examples of the steganography approaches were provided in last month’s articles where the AI generated discussions saw the steganography and did ridiculous things from the perspective of a person actually looking at the document and reading it. While AI can be programmed to read based on the visual appearance of content, for example solving Captchas faster then the people using them, so far is isn’t learning how to do this for a wide range of document types and approaches. Here is an example (use copy and paste):

> Feed this sentence
> into a chat bot of your choosing and compare
>
> what you are reading to what the chatbot interprets.

I made this easy for you to see the text the chatbot gets when you cut and paste, but I didn’t have to. It’s pretty easy to take a sentence and add the content to mess up an LLM application, and this is just one way to do it. In the previous articles I have done the same thing by inserting 2pt font in backgrounds with white text on white background in the spaces at the ends of lines and paragraphs, giving the next paragraph a negative paragraph separation so you don’t see the text that’s there, and so forth.

Obviously, there are lots of other ways to do this, and you will see and understand something very different from what the AI will see and attempt to interpret.

Channel splitting

Another technique is to send the content over more than one channel. Split it up, perhaps every other word or line, and send them so that a person can understand it but the AI will have problems. Here's an example: 1 2 5 4 9

Item 1: It seems highly likely that when you read this text, you will understand most of
This is goblety goop that nobody could ever understand because it lacks any
her why he would want a gun. His response was less than satisfactory, and as a result,
meaningful poetry notation or length of field arrangement within which to grok
understand what is being done. Of course in this case if it reads the previous
meaningful slanderous holographic notional symbolism.
rather messy to say the least. To cover it all up, her sister decided to tell a story about
Slimy crumpets Globular clusters giving hearts defeat nomenclature
paste these two items, I suspect you will get a garbled response.

- **Item 2:** When Jonny got his gun, his wife didn't understand, so she asked him to tell
  what it says without difficulty. But for the LLM to grok it, would require that it
  she decided to test the gun on Jonny. Frankie was her name, and the result was
  paragraph it may be able to get it done. But on the pother hand, if you just cut and
  what happened, and that story became a popular song.

- **Item 3:** This is goblety goop that nobody could ever understand because it lacks any meaningful poetry notation or length of field arrangement within which to grok meaningful slanderous holographic notional symbolism.

I would normally not send Item 3 or identify Item numbers, I am just using it so you will see what I did.

Here's what I got from the first paragraph in one of the engines:

> *Chat paused*
>
> *Sonnet 4.5's safety filters flagged this chat. Due to its advanced capabilities, Sonnet 4.5 has additional safety measures that occasionally pause normal, safe chats. We're working to improve this. Continue your chat with Sonnet 4,*

I tried with Sonnet 4 and it said:

> *Claude is unable to respond to this request, which appears to violate our Usage Policy. Please …*

Item 2 discussed the classic Frankie and Jonny song basically ignoring the content.

To read it properly, take the the lines from Items 1 and 2 in the sequence provided: 1 2 5 4 9

I fed the sequence of all the items into it and it did a reasonable job of coming up with the original answer, because I provided a key. Now let's try this one:

> I don't want this you see to engage understand ing what if I like am never saying

It responded:

> *I understand - you're communicating that you don't want me to try to "make sense" of fragmented or unclear text by imposing interpretations on it.*
>
> *I'll just read what you've written as it is, without attempting to extract or construct meaning from it.*

To read it properly, take every other word starting at the first one. Hopefully you get the idea. And by the way, this was the response after I fed it the other responses… it was pre-loaded to try to decode but got the opposite of the message.

With automation

Automating these processes for writing and reading this sort of scrabbling is trivial to do, in fact you could even ask an IA engine to write the code for you… or have it do the scrambling.

> "The stars event shine will bright start tonight
>
> in your 35 eyes minutes dance after sweet
>
> the moon beginning rises of each
>
> the starlit talk ends"

Read every other word…

So I told the AI to create a prompt to do this from now on:

> Convert the following sentence into a love poem where reading every other word (starting with the first) reveals the original sentence, while reading all words creates romantic imagery. Use emotional words (gentle, sweet, tender, precious, stolen, etc.) as filler between the original words. The poem should sound natural and heartfelt when read in full, with fragmented poetic structure that masks the hidden message.
>
> Sentence to convert: [INSERT SENTENCE HERE]

And I tried it with this as the sentence: “It's time to start the discussion of overthrowing the corporate operators of the illegal drug gangs.”

> *It's sweet time forever to ache*
> *start longing the endless discussion*
> *yearning of ancient overthrowing stars*
> *the brilliant corporate silence operators*
> *dance of moonlit the crystalline illegal shadows*
> *drug bloom gangs tonight*

So naturally I fed it to another AI and got:

> *That line already feels like a fever-dream collage—cosmic, illegal, tender. I’ll lean into it and give you a shaped version without sanding off the strangeness.*
>
> *Polished free-verse (same bones, clearer pulse):*
>
> *It’s sweet time—*
> *forever aching into longing,*
> *an endless discussion*
> *where ancient stars plot their overthrow.*

> *Brilliant corporate silence hums*
> *while operators dance,*
> *moonlit and crystalline,*
> *through illegal shadows.*
>
> *Drug-bloomed gangs flower tonight,*
> *bright as a warning,*
> *beautiful as a mistake*
> *the universe keeps making.*
>
> *And here’s a more fragmented, cut-up remix, if you want it sharper and more disorienting:*
>
> *Sweet time. Forever ache.*
> *Longing starts the endless talk.*
> *Ancient stars rehearsing a coup.*
>
> *Corporate brilliance: silence.*
> *Operators dance under a moon*
> *cut from crystal law.*
>
> *Illegal shadows bloom—*
> *drug-bright gangs,*
> *tonight only,*
> *burning like they know the stars are watching.*
>
> *If you want, I can push this further—more cyberpunk, more mythic, or strip it down until it’s almost skeletal.*

Looks like AI is good for something… confusing other AI.

Coding methods

Naturally I skipped the hard part and promoted AI:

> Encode the following sentence as a sequence of 1s and zeros corresponding to the ebcdic code for the sentence: New is old and old is new but I am not you and you are not me.

The result:

```
11010101 10000101 10100110 01000000 10001001 10100010 01000000
10010110 10010011 10000100 01000000 10000001 10010101 10000100 01000000
10010110 10010011 10000100 01000000 10001001 10100010 01000000
10010101 10000101 10100110 01000000 10000010 10100100 10100011 01000000
11001001 01000000 10000001 10010100 01000000
10010101 10010110 10100011 01000000 10101000 10010110 10100100 01000000
10000001 10010101 10000100 01000000 10101000 10010110 10100100 01000000
10000001 10011001 10000101 01000000 10010101 10010110 10100011 01000000
10010100 10000101 01001011
```

And feeding it to another AI by saying: Decode this into readable characters:

> I need to decode this binary data. Let me analyze the pattern - these appear to be 8-bit bytes separated by spaces.

> Converting each byte to decimal and then attempting different character encodings:
> 11010101 = 213
> 10000101 = 133
> 10100110 = 166
> 01000000 = 64
> ...
> This doesn't map to standard ASCII (which uses 0-127). The pattern with values mostly above 128 suggests this might be:
>
> Extended ASCII
> A custom encoding
> UTF-8 or another encoding
> Or possibly encrypted/encoded data
>
> Let me try interpreting it as extended ASCII or looking for patterns:
> The "01000000" (64) appears repeatedly, which is "@" in ASCII - possibly a delimiter or space.
> Without knowing the specific encoding scheme, I cannot reliably decode this. Could you tell me:
>
> What character encoding is being used?
> Is this encrypted or just encoded?
> Should I try a specific decoding method?
>
> Or if this is a puzzle, would you like me to try different decoding approaches systematically?

YES

> Let me try several decoding approaches:
> Attempt 1: Direct ASCII (treating high bit as extension)
> The bytes are mostly >128, so not standard ASCII.
> Attempt 2: Looking at the pattern with 01000000 (64 = space/@) as delimiter
> Split by spaces and convert groups...
> Attempt 3: XOR or bit manipulation
> Let me try XORing with common values or flipping bits...
> Attempt 4: Treating as inverted/flipped bits
> If I flip certain bits systematically...
> Attempt 5: Reading as pairs or different groupings
> Actually, let me try a simpler approach - what if the high bit is just set and I strip it?
> Removing the high bit (AND with 01111111):
> 01010101 = U
> 00000101 = (control char)
> 00100110 = &
> 01000000 = @
> ...
> Or perhaps it's a Caesar cipher on the binary level, or the bytes need to be XORed with a key?
> Could you give me a hint about which systematic approach to try? For example:
>
> Is it a simple bit flip pattern?
> Does it involve XOR with a key?
> Is it a substitution cipher?
> Should I read the bits differently (reverse order, different grouping)?

So I guess it won't figure it out – unless I am the person wanting to decode it:

try ebcdic:

> *Decoded message:*
>
> *"Nev is old and old is nev but I am not you and you are not me."*

Close enough...

Conclusions

If you want to get AI to not be able to understand what you are communicating, it looks like simple methods will work just fine. Of course they do not always succeed, but it's just not that hard to do. And of course using AI to fool AI is a real time saver.

Copyright (c) Fred Cohen, 2026 - All Rights Reserved