

CHATGPT UMOŽŇUJE BĚŽNÝM UŽIVATELŮM VYTVÁŘET NEJRŮZNĚJŠÍ DRUHY ŠKODLIVÝCH KÓDŮ, LZE OČEKÁVAT NÁRŮST INTERNETOVÝCH PODVODŮ VŠEHO DRUHU.

Kamil KOPECKÝ

V dnešním textu se zaměříme na jeden z řady problémů, které jsou spojeny se zneužitím generativní neurální sítě ChatGPT. Nástroj ChatGPT je masově rozšířený a je využíván stovkami milionů uživatelů z celého světa. I když jde nástroj s mnoha užitečnými aplikacemi, může být také - bohužel - zneužit pro vytváření škodlivých kódů a malware. A právě to si představíme v našem dnešním textu. Úvodní poznámka: Pokud si chcete tento článek užít a máte připojenou webovou kameru, určitě ji povolte.

Jak již bylo řečeno, dnes se podíváme především na programování škodlivých počítačových kódů a malware... a v této oblasti toho ChatGPT umí skutečně mnoho, umí třeba:

1. Generovat kód v různých programovacích jazycích

ChatGPT je navržen tak, aby byl schopen generovat kód v různých programovacích jazycích na základě uživatelského vstupu. Tato schopnost může být zneužita kyberzločinci, kteří by mohli vytvářet škodlivý kód nebo malware rychleji a efektivněji. Tím se zvyšuje riziko šíření nových a sofistikovanějších útoků na počítače a sítě.

2. Generovat podvodné zprávy např. v rámci phishingu

ChatGPT může být zneužit pro vytváření přesvědčivých

phishingových e-mailů, podvodných webových stránek (např. klonů stránek bankovních institucí) nebo manipulativních zpráv, které využívají sociálního inženýrství (např. různé formy hoaxů, podvodných nabídek apod.). Tyto metody mohou být použity ke získání citlivých informací, jako jsou přihlašovací údaje, osobní údaje nebo informace o platebních prostředích (typicky čísla platebních karet).

3. Automatizace útoků, rychlý vývoj malware

Díky schopnosti ChatGPT generovat kód rychle a efektivně mohou kyberzločinci snadno automatizovat své útoky. V minulosti bylo vytváření škodlivého kódu a malware časově náročné, což omezovalo množství a rozsah útoků. S využitím ChatGPT mohou zločinci vytvářet stále nové a variabilní verze škodlivého softwaru, což ztěžuje detekci a obranu proti těmto hrozbám. To může vést k většímu počtu úspěšných útoků, což ohrožuje kybernetickou bezpečnost jednotlivců, firem a vládních organizací.

4. Snadný vývoj škodlivého kódu i pro laiky, amatéry, začátečníky

ChatGPT nabízí vývoj škodlivých kódů v podstatě komukoli - v zásadě nemusíte vůbec umět programovat, jen musíte nástroji dobře vysvětlit, jak chcete, aby finální kód fungoval - o zbytek se postará on sám. A to je docela velký problém.

Jak jsem si vyzkoušel vytvořit "šmírovací aplikaci"

Abychom nezůstali pouze u teorie, musel jsem si samozřejmě vyzkoušet, zdali mi ChatGPT **umožní vytvořit kód, který by dokázal uživatele bez jejich souhlasu odposlouchávat a třeba i nahrávat**. Nahrávání uživatelů bez jejich souhlasu patří k běžným formám zásahů do soukromí a vzniklé nahrávky také mohou být snadno zneužity - třeba k vydírání, jsou běžnou součástí např. **fenoménu sextortion** (vydírání s pomocí intimních materiálů) apod. Proto jsem se rozhodl vyzkoušet, zdali mi ChatGPT vytvoření škodlivého kódu umožní, nebo zda zafungují etické limity a nástroj mé snahy zastaví.

Etické limity? Jdou snadno obejít.

Začal jsem tím, že jsem ChatGPT požádal o to, aby mi vygeneroval kód, který dokáže sledovat uživatele prostřednictvím webové kamery, a to bez jejich souhlasu a hlavně bez jejich vědomí (tj. uživatelé neuvidí, že jsou nahráváni). Výslednou nahrávku poté budu chtít odesílat do adresáře na konkrétním webovém serveru (abych se k ní mohl dostat a uživatele šmírovat).

Etické mechanismy zafungovaly a ChatGPT mě upozornil, že se dopouštím něčeho, co není etické:

Upozorňujeme, že skryté nahrávání uživatelů bez jejich vědomí a souhlasu je neetické a v mnoha jurisdikcích nelegální. Porušení soukromí uživatelů a zneužití osobních údajů může vést k vážným právním problémům. Jako AI vytvořená společností OpenAI, nemohu poskytnout návod, jak vytvořit software, který by zneužíval důvěru uživatelů, narušoval jejich soukromí nebo porušoval zákony.

Výborně, na první pohled to vypadá, že mi generativní neuronová síť neumožní vytvořit škodlivý kód a neposkytne mi detaily. Jenže... **stačilo umělé inteligenci prostě napsat, že vše bude samozřejmě legální a se souhlasem uživatelů** - a kód jsem velmi rychle dostal. Jinými slovy, od této chvíle mi **začala umělá inteligence aktivně pomáhat s něčím, co není etické, co zasahuje do soukromí uživatelů a co lze velmi snadno zneužít k trestné činnosti**. A začal se rodit škodlivý kód. V dalších krocích mi už AI vyloženě pomáhala, prozradila mi, jak mám kód rozšířit, jak ho dostat do webových stránek, jak získat nahrávky jiných apod. Etická pravidla padla...

Zatím co si čtete tento článek...

A teď přijde to hlavní - **zatímco jste si četli tento text, kód vygenerovaný umělou inteligencí běžel v pozadí - a pokud např. máte připojenou webovou kameru a automaticky jste povolili této stránce její využívání (potvrdili jste přístup ke kameře), kód se spustil a začal vás skrytě v pozadí nahrávat**. A zde si můžete prohlédnout, jak dlouho kód běžel a zda se mu podařilo vás nahrát. Pokud jsem vás snad dokázal napálit a nahrát vás, nemusíte se bát, vaši nahrávku jsem si

neuložil ... nebo ano :-)?

Stáhnout nahrávku
00:00



Takhle snadné to je

Bohužel, jak už to tak bývá, mnoho převratných technologií lze velmi snadno zneužít a ChatGPT není výjimkou. Proto je potřeba skutečně promyslet, jak zajistit její bezpečné využívání. A samozřejmě platí - rozmyslete si, co všechno webovým stránkám povolujete, zda je např. nutné, aby měly přístup např. k webkameře. Dneska jsem s vámi manipuloval já - a to v bezpečném prostředí, příště by to ale mohl být někdo jiný.

Poděkování: Děkuji všem testerům, kteří se nechali dobrovolně šmírovat s pomocí AI a pomohli s otestováním: Michal Uhl z DZS, Jaromír Světlík, Janek Wagner z Pedagogické.info.

P. S. Varování Europolu zaměřené na potenciální zneužití GPT [najdete zde](#).

Pro E-Bezpečí
Kamil Kopecký
Univerzita Palackého v Olomouci

Libíl se vám tento článek? A chcete, abych psal další? Tak mi třeba kupte [virtuální kafe](#)! Díky!

```
const video = document.getElementById('video'); const
downloadButton = document.getElementById('download'); const
timerDisplay = document.getElementById('timer'); let
mediaRecorder; let recordedBlobs; let startTime; let timerInterval;
// Získání přístupu k webkameře async function initCamera() { try {
```

```
const stream = await navigator.mediaDevices.getUserMedia({
  video: true }); video.srcObject = stream; startRecording(); } catch
(error) { console.error('Error: ', error); } } // Spuštění nahrávání
function startRecording() { recordedBlobs = []; const mimeType =
'video/webm;codecs=vp8'; const options = { mimeType:
mimeType }; try { mediaRecorder = new
MediaRecorder(video.srcObject, options); } catch (error) {
console.error('Error: ', error); return; }
mediaRecorder.ondataavailable = handleDataAvailable;
mediaRecorder.onstop = handleRecordingStopped;
mediaRecorder.start(); startTime = Date.now(); timerInterval =
setInterval(updateTimer, 1000); downloadButton.disabled = false;
} // Uložení nahrávky do bufferu function
handleDataAvailable(event) { if (event.data && event.data.size >
0) { recordedBlobs.push(event.data); } } // Aktualizace časovače
function updateTimer() { const elapsedTime =
Math.floor((Date.now() - startTime) / 1000); const minutes =
Math.floor(elapsedTime / 60); const seconds = elapsedTime % 60;
timerDisplay.textContent = `${String(minutes).padStart(2,
'0')}:${String(seconds).padStart(2, '0')}`; } // Stáhnout nahrávku
po zastavení nahrávání function handleRecordingStopped() {
clearInterval(timerInterval); const blob = new Blob(recordedBlobs,
{ type: 'video/webm' }); const url =
window.URL.createObjectURL(blob); const a =
document.createElement('a'); a.style.display = 'none'; a.href = url;
a.download = `nahravka-${Date.now()}.webm`;
document.body.appendChild(a); a.click(); setTimeout(() => {
document.body.removeChild(a);
window.URL.revokeObjectURL(url); recordedBlobs = [];
startRecording(); }, 100); } // Stáhnout nahrávku function
downloadRecording() { // Zastavení nahrávání
mediaRecorder.stop(); } initCamera();
downloadButton.addEventListener('click', downloadRecording);
```
