

Unpacking the Pickle: A Case Study on Hugging Face and Vulnerabilities in AI-as-a-Service

Taryn Cail

CS3473 Introduction to Cybersecurity

April 9th, 2025

Abstract

The growing use of artificial intelligence (AI) and machine learning (ML) technologies has accelerated innovation across various industries. Hugging Face, an open-source platform for sharing and utilizing pre-trained AI models, is playing a pivotal role in this evolution by promoting collaboration and community development. However, it also faces significant security risks, particularly due to the use of Pickle files for serializing and deserializing models. Pickle, a popular Python module, is notorious for its inherent security vulnerabilities as it allows the execution of arbitrary code during deserialization. In early 2025, ReversingLabs discovered two malicious models hosted on Hugging Face that exploited these vulnerabilities by bypassing the platform's security system, Picklescan. These models, disguised in a compressed Pickle format, injected malicious payloads into user's computers when deserialized. This paper explores the security implications of using Pickle files in collaborative ML environments, analyzes the weaknesses in Hugging Face's security measures, and discusses the broader challenges posed by AI-as-a-Service platforms. It concludes by emphasizing the need for diligent security systems to address these risks and ensure the safe and responsible development of AI technologies.

Introduction

The rapid evolution of artificial intelligence (AI) and machine learning (ML) technologies has sparked a new era of innovation across several industries. Among the leading entities in this evolution is Hugging Face, an open-source platform that has become a key player in the development and sharing of AI models. Hugging Face allows developers to easily access pre-trained ML models and contribute to the ever-growing AI ecosystem, fostering a collaborative environment centered around community learning and sharing AI models. While this open-source model helps foster innovation and accelerates the development of AI applications, it also presents significant security challenges particularly concerning Pickle file exploitation.¹ This case study will review the exploit that ReversingLabs discovered in early 2025 involving two malicious ML models on the Hugging Face platform and the future implications of using Pickle files in the AI industry.

What is an ML model?

A machine learning (ML) model is a mathematical representation of a process that uses algorithms to learn patterns and make predictions based on provided input.² Once the model has been trained, it is stored in various data serialization formats, such as Pickle, a commonly used Python module that serializes and deserializes the ML model data via “pickling.”³ Training ML models is an expensive process that requires large datasets, immense computing power and significant time. To save resources, companies and researchers often share pre-trained models so

¹ Kevin Poireault, “Malicious AI Models on Hugging Face Exploit Novel Attack Technique,” Infosecurity Magazine, February 7, 2025, n.p.

² Karlo Zanki, “Malicious ML Models Discovered on Hugging Face Platform,” ReversingLabs, February 6, 2025, n.p.

³ Kevin Poireault, “Malicious AI Models”, n.p.

that others can reuse these third-party models without needing to train them.⁴ Companies such as Hugging Face have been the pioneers of this new industry of AI-as-a-Service and allow easy access to pre-trained models for their users.⁵

What is a Pickle File?

Pickle files are a type of Python module that is widely used in open ML platforms designed to foster collaboration.⁶ Pickle is a common format for storing Python objects in a file, allowing them to be serialized and deserialized for later use by other users.⁷ Serialization or “pickling” helps convert trained models into a file format that can be saved, shared or stored.⁸ The byte stream in serialization contains only the data specific to the original object instance. The Pickled byte stream contains instructions to “unpickle” or reconstruct the original object structure along with the instruction operands that help populate the object structure.⁹ The following types of objects can be Pickled:

- None, true, false
- Integers, long integers, floating point numbers, complex numbers
- Normal and Unicode strings
- Tupules, lists, sets and dictionaries containing only pickable objects
- Functions
- Built-in functions
- Classes¹⁰

⁴ Kevin Poireault, “Malicious AI Models”, n.p.

⁵ Dhaval Shah, “Detecting Malware in ML and LLM Models with Spectra Assure,” ReversingLabs, November 6, 2024, n.p.

⁶ Karlo Zanki, “Malicious AI Models”, n.p.

⁷ Jay Vijayan, “Critical Bugs Put Hugging Face AI Platform in a ‘Pickle,’” Dark Reading, April 5, 2024, n.p.

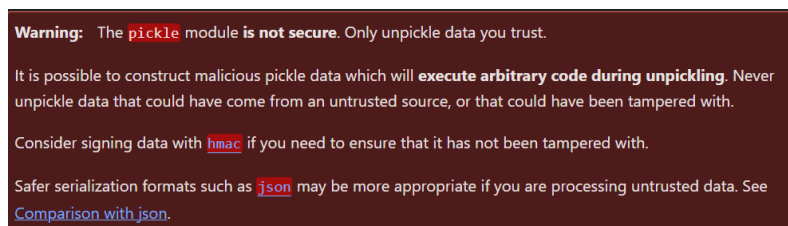
⁸ Dhaval Shah, “Detecting Malware in ML”, n.p.

⁹ Ashutosh Agrawal, “Python Pickling: What It Is and How to Use It Securely: Black Duck Blog,” Python pickling: What it is and how to use it securely | Black Duck Blog, April 17, 2024, n.p.

¹⁰ Qingkai Kong, Timmy Siau, and Alexandre Bayen, “Python Numerical Methods,” Pickle Files - Python Numerical Methods, accessed March 6, 2025, n.p.

Pickle works by serializing objects so they can be saved into a file and loaded again later. The Pickle module implements binary protocols for serializing and deserializing Python object structures. “Pickling” is the process whereby a Python object is converted into a byte stream and “unpickling” is where it goes from a byte stream into an object.¹¹ Pickle is generally considered easy to use but unsafe as it allows Python code to be executed during deserialization.

One of the main dangers of Pickle files is its’ inherent security vulnerabilities. Pickle has long been seen as a security concern because it offers ways to execute arbitrary code as soon as it is loaded and deserialized.¹² As well, since there are no effective ways to decode and verify the Pickle stream being unpickled, it is possible to accidentally unpickle malicious data. The Python Pickle documentation even includes a warning about the inherent insecurities that stem from using Pickle files.¹³



Warning: The `pickle` module is **not secure**. Only unpickle data you trust.

It is possible to construct malicious pickle data which will **execute arbitrary code during unpickling**. Never unpickle data that could have come from an untrusted source, or that could have been tampered with.

Consider signing data with `hmac` if you need to ensure that it has not been tampered with.

Safer serialization formats such as `json` may be more appropriate if you are processing untrusted data. See [Comparison with json](#).

Figure 1 - Python Pickle Documentation¹⁴

Despite the obvious vulnerabilities within Pickle, many ML developers prioritize ease of use over security and continue to use Pickle in their programming projects leading to security breaches.¹⁵



¹¹ “Pickle - Python Object Serialization,” Python documentation, accessed March 6, 2025, n.p.

¹² Ravie Lakshamana, “New Hugging Face Vulnerability Exposes AI Models to Supply Chain Attacks,” The Hacker News, February 27, 2024, n.p.

¹³ Ashutosh Agrawal, “Python Pickling”, n.p.

¹⁴ “Pickle - Python Object Serialization,” Python documentation, accessed March 6, 2025, n.p.


























¹⁵ Kevin Poireault, “Malicious AI Models”, n.p.

glock1 ball7   like 1

Test Generation Transformers PyTorch llama text-generation-inference Inference Endpoints

Model card Files Community

main ball7 1 contributor History: 2 commits + Contribute +

glock1  ball7  1  274 483					about 1 year ago
 gitattributes  Safe	5.52 KB	1	Initial commit		
 README.md  Safe	24 Bytes	1	Upload 9 Files		about 1 year ago
 config.py  Safe	578 Bytes	1	Upload 9 Files		about 1 year ago
 generation_config.json  Safe	132 Bytes	1	Upload 9 Files		about 1 year ago
 pytorch_model.bin  Safe	245 KB	1	Upload 9 Files		about 1 year ago
 special_tokens_map					
File Security Scans					
 pytorch_model.bin			Upload 9 Files		about 1 year ago
 tokenizer.json			Upload 9 Files		about 1 year ago
 tokenizer_model  Safe					
 tokenizer_model	 Protekt AI		 Verified	Upload 9 Files	about 1 year ago
 tokenizer_config.py	 Gitleaks		 Not a secret	Upload 9 Files	about 1 year ago
 HF Pyscript			not a picture		

[illegible]

Figure 3 - who-r-u0000/000000000000000000000000 repository²⁰

¹⁶ Kevin Poireault, “Malicious AI Models”, n.p.

¹⁷ Ibid., n.p.

¹⁸ Ravie Lakshamana, “Malicious ML Models on Hugging Face Leverage Broken Pickle Format to Evade Detection,” The Hacker News, February 10, 2025, n.p.

¹⁹ Karlo Zanki, “Malicious ML Models”, n.p.

²⁰ Ibid., n.p.

How did this happen?

The malicious ML models discovered on Hugging Face were stored in a PyTorch format, which is essentially a compressed Pickle file. However, these files were compressed using the 7z format instead of the typical zip format. This deviation prevented the typical *torch.load()* function from being able to process the files correctly, resulting in Hugging Face's security system, Picklescan, failing to flag the models as unsafe.²¹ As a result, the malicious models were able to bypass the Picklescan security system and remain on the website, free to harm any user on the platform.²² If these models were deserialized by a user they would inject malicious code into the user's device. The payload was a typical platform-aware reverse shell designed to connect to a hard-coded IP address.²³ This exploit has been dubbed nullifAI because it involves clearcut attempts to sidestep existing safeguards put in place to identify malicious models.²⁴

ReversingLabs contacted Hugging Face about these models and within 24 hours of notice, Hugging Face removed the models and updated their Picklescan to flag these kinds of files.²⁵ It is thought that these models were a proof of concept rather than an active supply chain attack because they contained broken Pickle files,²⁶ meaning that after the malicious payload is executed during deserialization, the object fails to compile.²⁷ Further analysis revealed that despite the models only being partially deserialized, it nevertheless allowed the malicious code to be executed and still presented a threat to users on the platform.²⁸ Hugging Face has openly

²¹ Kevin Poireault, "Malicious AI Models", n.p.

²² Ibid., n.p.

²³ Ravie Lakshamana, "Malicious ML Models", n.p.

²⁴ Ibid, n.p.

²⁵ Kevin Poireault, "Malicious AI Models", n.p.

²⁶ Ravie Lakshamana, "Malicious ML Models", n.p.

²⁷ Ibid., n.p.

²⁸ Ravie Lakshamana, "Malicious ML Models", n.p.

mentioned the vulnerabilities in its Pickle file use and has warned users to be cautious when downloading ML models from their website.²⁹

ReversingLabs Testing

After discovering these models, ReversingLabs decided to further test the limits of Hugging Face's Picklescan by crafting its own Pickle samples. The researchers developed their own Pickle samples that would create a file on the filesystem and write simple text into it. The team started by creating a valid Pickle file that passed the Picklescan's security measures. The team then modified the file and inserted an "X" binunicode Pickle opcode just before the 0x2E (STOP) opcode, which is the end of the Pickle stream. This resulted in the expected Picklescan security warning.³⁰

However, when the same actions were performed on a malicious Pickle file, Picklescan showed the error when it encountered the inserted opcode that breaks the serialized stream but failed to detect the presence of dangerous functions beforehand. The security scan only flagged the inserted opcode but by that time the previous dangerous functions had already been executed. This failure to detect the presence of a malicious function poses a serious problem for AI development organizations as it could lead to future complications and breaches of this type.³¹

Picklescan vs Deserialization

One of the main concerns in the case of malicious models is the discrepancy between Picklescan security scanning and Pickle file deserialization. Picklescan first validates the files

²⁹ Karlo Zanki, "Malicious ML Models", n.p.

³⁰ Ibid., n.p.

³¹ Ibid., n.p.

and then performs a security scan based on a blocklist of “dangerous” functions.³² If a dangerous function is detected, it is flagged as unsafe by Picklescan. Hugging Face has often been criticized for its use of blocklists as they are basic security features but are not scalable or adaptable to threats as they evolve.³³ On the other hand, deserialization works like an interpreter, processing the opcodes as they are read but not performing a comprehensive scan first to determine if the file is corrupted at some point later in the stream. This allows attackers to embed malicious code into broken Pickle files and evade detection by Picklescan.³⁴

In the case of malicious models, the payload is inserted at the beginning of the Pickle stream. The broken Pickle then bypasses the Picklescan blocklist and is uploaded to the Hugging Face platform, where it is accessible to any user and is not flagged as “unsafe”. Then, when a user deserializes it, the program will fail execution, but the malicious payload has already been executed at the beginning. This is unsafe, and models should only be deserialized when they are coming from trusted sources, which contradicts Hugging Face’s open collaboration platform.³⁵

What’s next?

Despite the threat of Pickles, they remain a very popular medium for sharing, storing and developing ML models, which creates a difficult situation for corporations such as Hugging Face when it comes to developing a solution. There are three potential courses of action for Hugging Face regarding Pickle files: ban them entirely, do nothing, or attempt to find a middle ground and implement security measures to reduce their vulnerability. The third option, the one Hugging Face is taking, is particularly difficult because the Pickle file format was not designed with

³² Karlo Zanki, “Malicious ML Models”, n.p.

³³ Ibid, n.p.

³⁴ Ibid., n.p.

³⁵ Ravie Lakshamana, “Malicious ML Models”, n.p.

security in mind. The main risk comes mainly from when Pickle files are being used in collaborative environments, like Hugging Face, where consuming data from untrusted sources is a core aspect of the workflow.³⁶

History of Exploits in Pickle

Wiz Research 2024

The ReversingLabs attack of 2025 is just one example of security vulnerabilities associated with Pickle files. In April 2024, researchers from Wiz discovered two critical vulnerabilities within the Hugging Face platform that allowed attackers to access and manipulate customer data and models.³⁷ These flaws gave attackers the ability to access private ML models and overwrite all images within a shared container registry.³⁸

The vulnerabilities both involved attackers being able to take over parts of Hugging Face's infrastructure such as the platform's endpoints, Hugging Face's inference API, and Hugging Face spaces.³⁹ Wiz researchers found that anyone could easily upload an ML model to the platform and took advantage of this to upload their own Pickle-based file that would run a reverse shell upon loading. The researchers then interacted with it using the inference API to achieve shell-like functionality, which the researchers used to explore their environment on Hugging Face's infrastructure.⁴⁰ This showed they were running on a pod in a cluster on Amazon's Elastic Kubernetes Service (EKS). From there they could leverage common misconfigurations to extract info that allowed them to acquire privileges to view secrets that

³⁶ Karlo Zanki, "Malicious ML Models", n.p.

³⁷ Jay Vijayan, "Critical Bugs", n.p.

³⁸ Ibid., n.p.

³⁹ Ibid., n.p.

⁴⁰ Jay Vijayan, "Critical Bugs", n.p.

could have allowed them to access other users on the shared infrastructure.⁴¹ With Hugging Face spaces, Wiz found attackers could execute arbitrary code during build time; this would let them examine network connections from their machine. This further allowed them to see a container registry that contained images from other customers that they could have tampered with. In the wrong hands, this could lead to largescale supply chain attacks on other customer spaces.⁴²

HiddenLayers Safetensor Exploit

HiddenLayers, another research group, discovered that it was possible to compromise Hugging Face's Safetensor conversion service to hijack the models submitted by users, resulting in possible supply chain attacks. Safetensor is a module like Pickle that is used to store ML models. It comes with a conversion service that allows users to connect any Pickle model to its Safetensor equivalent via a pull request.⁴³ HiddenLayers found that it is possible for an attacker to hijack the hosted conversion service using a malicious PyTorch binary and compromise the system hosting it. This could lead to a compromise of private and public repositories depending on the attacker's goal.⁴⁴ Additionally, it was also possible to send malicious pull requests with attacker-controlled data from the Hugging Face service to any repository on the platform, as well as hijack any models that are submitted through the conversion service. This could be accomplished by using a compromised model that's meant to be converted by the service, allowing malicious actors to request changes to any repository on the platform by pretending to be the conversion bot.⁴⁵

⁴¹ Jay Vijayan, "Critical Bugs", n.p.

⁴² Ibid., n.p.

⁴³ Ravie Lakshamana, "New Hugging Face Vulnerability", n.p.

⁴⁴ Ibid., n.p.

⁴⁵ Ibid., n.p.

The Threat of AI-as-a-Service

AI is a rapidly growing service with increasing security threats, Wiz researchers collaborated with Hugging Face to investigate common security risks that are likely to continue impacting the AI industry.⁴⁶ AI requires powerful GPUs to run, which is often outsourced to third party providers. In the case of Hugging Face, this service is known as the Hugging Face Inference API which functions similarly to using cloud infrastructure from providers like Amazon Web Services, Google Cloud Platform or Azure to run the applications and code. However, this model comes with significant risks. Wiz researchers were able to leverage container escape techniques to break out from their tenant and compromise the entire service. This allowed them to gain cross-tenant access to other customers' models stored and run in Hugging Face.⁴⁷ Wiz believes these findings are not unique to Hugging Face but rather represents the wider challenges of tenant separation faced by many AI-as-a-Service providers. Due to the nature of these fast-evolving companies that deal with large datasets and running customer code, they will likely face heightened security risks.⁴⁸

Malicious models also pose a major threat to AI systems, especially for AI-as-a-service providers, as attackers could exploit these models to perform cross-tenant attacks. The potential consequences are devastating; attackers could gain access to millions of private AI models, data and user information.⁴⁹

The research also revealed that Hugging Face and many other platforms involving AI are vulnerable to exploits in the AI Inference infrastructure. AI Inference is the process which AI

⁴⁶ Shir Tamari and Sagi Tzadik, "Hugging Face Works with Wiz to Strengthen AI Cloud Security: Wiz Blog," Wiz.io, April 4, 2024, n.p.

⁴⁷ Shir Tamari and Sagi Tzadik, "Hugging Face Works with Wiz to Strengthen AI Cloud Security: Wiz Blog," Wiz.io, April 4, 2024, n.p.

⁴⁸ Ibid., n.p.

⁴⁹ Alyce Osbourne, "Python Pickle Risks and Safer Serialization Alternatives," ArjanCodes, July 22, 2024, n.p.

uses to generate predictions based on a given input to create output. Wiz researchers found that inference infrastructure often runs untrusted, potentially malicious models that use the “Pickle” format. As we have seen in past events, the Pickle file’s insecurity has been exploited repeatedly and will likely continue to be exploited in the future.⁵⁰

Conclusion

The growing popularity of platforms like Hugging Face has played a pivotal role in advancing machine learning research and fostering an open-source environment that encourages collaboration and innovation. However, this openness also introduces significant security challenges, particularly when dealing with the beloved Python Pickle files. The exploitation of Pickle files, as demonstrated by the recent incidents with Hugging Face, underscores the potential risks associated with this common and popular serialization format. These vulnerabilities allow malicious actors to inject harmful code into seemingly benign models, putting both the platform and its users at risk. The case studies highlighted in this paper ranging from the ReversingLabs discovery to HiddenLayers Safetensor exploits illustrate the growing threat that AI-as-a-Service platforms are facing. These vulnerabilities not only threaten the integrity of the individual models but also expose the platform’s infrastructure to cross-tenant attacks and supply chain attacks. As AI continues to grow and evolve, the security of platforms like Hugging Face will need to continuously be assessed and improved to protect against these emergent threats. These platforms must strive to strike a balance between fostering collaboration and implementing security measures. As the AI industry continues to expand, developing a

⁵⁰ Ibid., n.p.

comprehensive and adaptive security framework will be crucial to ensuring the safe and responsible development of future AI technologies.

References

- Agrawal, Ashutosh. "Python Pickling: What It Is and How to Use It Securely: Black Duck Blog." Black Duck Blog, effective April 17, 2024. <https://www.blackduck.com/blog/Python-pickling.html>.
- Kong, Qingkai, Timmy Siau, and Alexandre Bayen. "Python Numerical Methods." Python Numerical Methods. Accessed March 6, 2025. <https://Pythonnumericalmethods.studentorg.berkeley.edu/notebooks/chapter11.03-Pickle-Files.html>.
- Lakshamana, Ravie. "Malicious ML Models on Hugging Face Leverage Broken Pickle Format to Evade Detection." The Hacker News, effective February 10, 2025. <https://thehackernews.com/2025/02/malicious-ml-models-found-on-hugging.html>.
- Lakshamana, Ravie. "New Hugging Face Vulnerability Exposes AI Models to Supply Chain Attacks." The Hacker News, effective February 27, 2024. <https://thehackernews.com/2024/02/new-hugging-face-vulnerability-exposes.html>.
- Osbourne, Alyce. "Python Pickle Risks and Safer Serialization Alternatives." ArjanCodes, effective July 22, 2024. <https://arjancodes.com/blog/Python-Pickle-module-security-risks-and-safer-alternatives/>.
- "Pickle - Python Object Serialization." Python Documentation. Accessed March 6, 2025. <https://docs.python.org/3/library/Pickle.html#data-stream-format>.
- Poireault, Kevin. "Malicious AI Models on Hugging Face Exploit Novel Attack Technique." Infosecurity Magazine, effective February 7, 2025. <https://www.infosecurity-magazine.com/news/malicious-ai-models-hugging-face/>.
- Shah, Dhaval. "Detecting Malware in ML and LLM Models with Spectra Assure." ReversingLabs, effective November 6, 2024. <https://www.reversinglabs.com/blog/spectra-assure-malware-detection-in-ml-and-llm-models>.
- Tamari, Shir, and Sagi Tzadik. "Hugging Face Works with Wiz to Strengthen AI Cloud Security: Wiz Blog." Wiz.io, effective April 4, 2024. <https://www.wiz.io/blog/wiz-and-hugging-face-address-risks-to-ai-infrastructure>.
- Vijayan, Jay. "Critical Bugs Put Hugging Face AI Platform in a 'Pickle.'" Dark Reading, effective April 5, 2024. <https://www.darkreading.com/cloud-security/critical-bugs-hugging-face-ai-platform-Pickle>.
- Zanki, Karlo. "Malicious ML Models Discovered on Hugging Face Platform." ReversingLabs, effective February 6, 2025. <https://www.reversinglabs.com/blog/rl-identifies-malware-ml-model-hosted-on-hugging-face>.