01 · THE THREAT

MODEL STEALING

When Your AI Becomes Someone Else's Product

A practical security analysis of model extraction, intellectual property exposure and the risks of exposing AI through APIs.

You built the model. Someone else may be learning to reproduce it.

Imagine spending months developing an AI model. You invest in data, engineering, infrastructure, training and testing. Eventually, you make it available to customers through an API.

Customers send requests. Your model processes them and returns answers.

But there is a security question that is easy to overlook:

What prevents someone from systematically collecting those answers and using them to build a competing model?

This is the basic idea behind a model stealing attack, also known as model extraction.

An attacker interacts with a model, collects its responses and uses those input-output examples to train another model that imitates its behaviour.

The attacker does not necessarily need access to the original source code, training data or model weights.

They may only need access to the model's interface.

WHY BUSINESSES SHOULD CARE

For companies whose competitive advantage depends on proprietary AI, the model itself can represent substantial intellectual property.

Its value may come from:

- A specialised prediction capability.
- A proprietary classification or recommendation system.
- Fine-tuning and domain-specific behaviour.
- Expensive training and development.
- A model that supports a unique commercial product.

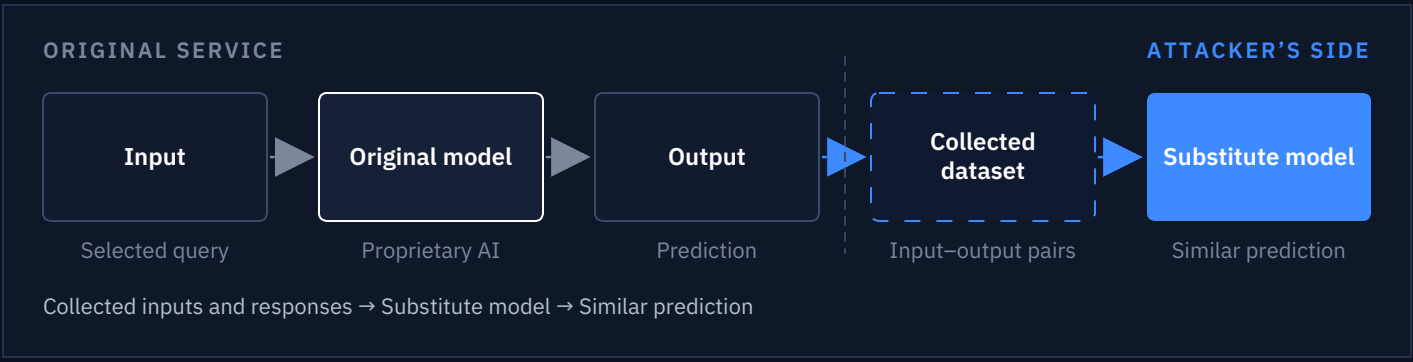
If an attacker can reproduce enough of that functionality, the company may lose part of the advantage it spent time and money creating.

The uncomfortable reality: an AI interface designed to serve customers can also become a source of information for someone trying to reproduce the underlying system.

02 · HOW MODEL STEALING WORKS

The model answers. The attacker learns.

A model extraction attack can be understood in five stages.



1 **Access**

The attacker obtains access to a model through a public API, application or another authorised-looking interface.

2 **Querying**

The attacker sends carefully selected inputs. These are not necessarily random questions. They are designed to reveal how the model behaves across different situations.

3 **Collection**

The attacker records the relationship between each input and the model's output.

4 **Training**

The collected examples become training material for a separate model, often called a substitute or student model. The substitute is trained to reproduce the original model's responses.

5 **Approximation**

The attacker evaluates how closely the substitute behaves like the original. If the substitute achieves sufficient agreement on the tasks that matter, it may reproduce part of the original model's commercial functionality.

EXAMPLE · STAGE 3

INPUT

Customer transaction characteristics.

ORIGINAL MODEL

Fraud risk score.

COLLECTED RESULT

The score returned by the model.

Repeated across many carefully selected examples, these observations can reveal useful patterns.

TECHNICAL DISTINCTION

Model extraction does not automatically produce an identical copy. A substitute model may imitate the original on a limited set of tasks while having different architecture, parameters, training data and capabilities.

The result depends on query access, model complexity, output information, available data and the attacker's resources.

The objective is often functional imitation, not necessarily recovery of the original model files.

03 · A REALISTIC BUSINESS SCENARIO

The exposed prediction API

Consider a fictional company called **FinPredict**.

FinPredict develops a proprietary AI model that estimates credit risk for financial institutions. Its customers access the model through an API.

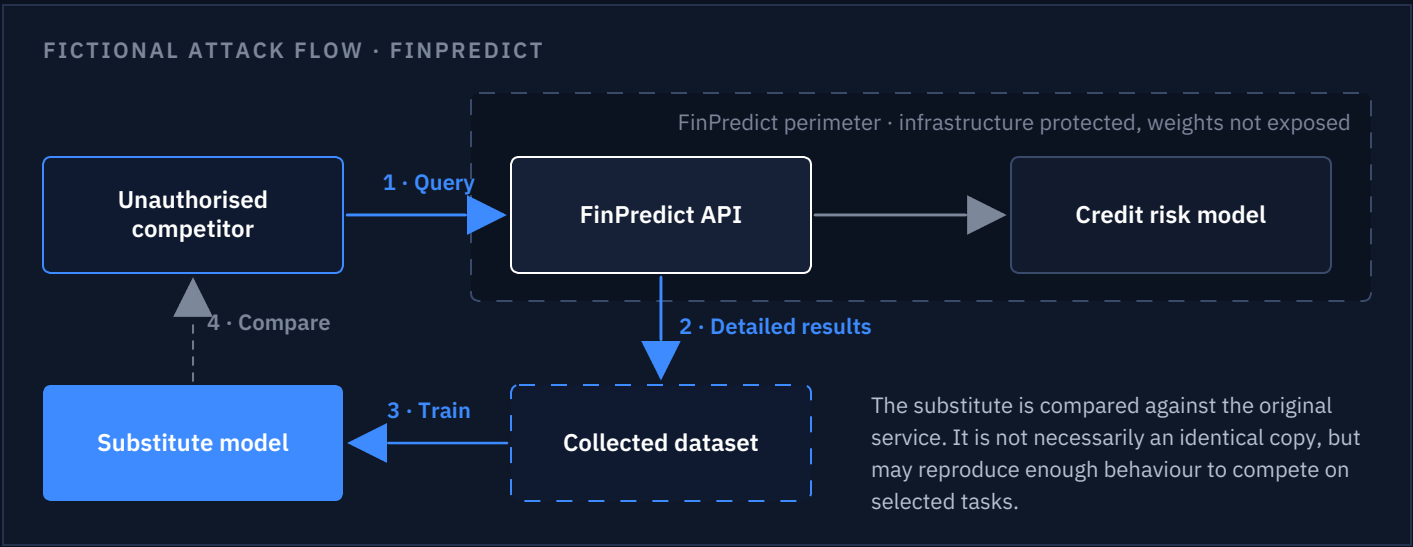
The company protects its infrastructure and does not expose its model weights.

However, the API accepts prediction requests and returns detailed results.

An unauthorised competitor obtains access to the service and begins submitting large numbers of carefully selected requests.

Over time, it collects a dataset containing:

- Submitted input characteristics.
- Model predictions.
- Confidence or probability information, where exposed.
- Repeated observations across different input combinations.



Where did the security failure occur?

The company may have protected its model files while overlooking the information exposed through its prediction interface. Potential weaknesses include:

- Weak API authentication.
- Excessive query access.
- No effective usage monitoring.
- No detection of systematic extraction patterns.
- Unnecessary exposure of confidence scores or detailed outputs.
- Insufficient separation between customer access and internal model operations.

THE LESSON

Protecting model files is not enough.
A model can be exposed indirectly through the information it returns.

This scenario is illustrative. It does not describe a specific real-world victim or claim that every accessible AI API is vulnerable.

04 · BUSINESS IMPACT

What can an attacker gain?

Model stealing is an intellectual property risk, but the consequences can go further.

01

Loss of competitive advantage

A substitute model may allow another organisation to reproduce valuable functionality without making the same investment in development.

02

Unauthorised commercial use

A copied or approximated model may be integrated into another product or service without the original developer's permission.

03

Reduced control over proprietary technology

Once a model's behaviour has been reproduced, the original organisation may have less control over how that capability is distributed or used.

04

Lower-cost experimentation against the model

A sufficiently accurate substitute can allow an attacker to experiment locally rather than repeatedly querying the original service. Depending on the fidelity of the substitute, this may help identify weaknesses in its behaviour.

05 Potential privacy implications

Related, but distinct attack classes.

MODEL EXTRACTION

Aims to reproduce model behaviour.

≠

MEMBERSHIP INFERENCE · MODEL INVERSION

Aim to infer information about training data.

In some circumstances, access to a model or its outputs may contribute to additional privacy risks. However, extracting a substitute model does not automatically reveal the original training dataset.

Who is exposed?

The risk is particularly relevant when the model represents a core part of the company's intellectual property.

ORGANISATIONS THAT SHOULD CONSIDER THIS THREAT

- AI startups with proprietary models
- FinTech companies offering prediction APIs
- Healthcare tech using specialised models
- SaaS businesses integrating proprietary AI
- Enterprises building internal ML services
- Companies selling models as a service

05 · DEFENCES

How to reduce the risk

You cannot protect what you do not monitor.



There is no single control that eliminates model extraction risk. Effective protection requires multiple layers.

Rate limiting alone is not sufficient: extraction can occur slowly over time.

01 Strong API authentication

Identify who is accessing the model. Use appropriate authentication, authorisation and credential management.

02 Query limits and usage controls

Apply rate limits, quotas and usage policies. Restrict unusually high-volume or repetitive querying.

03 Behavioural monitoring

Monitor query patterns rather than only counting requests. Look for unusual activity such as:

- Large volumes of systematically varied inputs.
- Repeated probing of decision boundaries.
- Unusual usage outside normal customer behaviour.
- Repeated requests designed to obtain detailed outputs.

04 Minimise exposed information

Return only what customers need. For example, avoid exposing confidence scores, logits, internal metadata or detailed prediction information when they are not necessary for the product. This reduces the amount of information available to someone attempting extraction.

05 Protect model infrastructure

Secure model registries, storage, checkpoints, deployment environments and internal repositories. Use least-privilege access, encryption where appropriate, secrets management and audit logging.

06 Monitor costs and anomalies

Unusual inference consumption can be a useful warning signal. Monitor usage costs, request frequency, account behaviour and unexpected changes in traffic.

07 Consider model watermarking and fingerprinting

Watermarking and behavioural fingerprinting may help identify unauthorised reuse in certain contexts. These methods have limitations and should not be treated as guaranteed proof of theft.

A PRACTICAL PRINCIPLE

Security should cover both sides of the model.

At rest
Weights, checkpoints, repositories and infrastructure.

In use
APIs, outputs, queries, accounts and inference behaviour.

06 · SECURITY ASSESSMENT

What security teams should test

A model can be accessible without being adequately protected.

A security assessment should examine more than whether an API requires an authentication token. It should evaluate how the model behaves under realistic and authorised usage scenarios. At LINK TECHREATIONS, we consider the following assessment areas relevant to model extraction risk.

ASSESSMENT AREA	CHECKPOINTS
A1 **API exposure**	☐ Can unauthorised users access prediction endpoints? ☐ Are authentication and authorisation correctly enforced? ☐ Can access controls be bypassed?
A2 **Query behaviour**	☐ Are query limits enforced consistently? ☐ Can an account generate unusually large volumes of requests? ☐ Can repeated requests be used to probe model behaviour?
A3 **Output exposure**	☐ Does the API expose more information than necessary? ☐ Are confidence values, probabilities or internal metadata available? ☐ Could output detail make model imitation easier?
A4 **Monitoring and detection**	☐ Are suspicious query patterns logged? ☐ Can unusual usage be detected? ☐ Are alerts and response procedures defined?
A5 **Model infrastructure**	☐ Are model weights and checkpoints protected? ☐ Are model registries and repositories access-controlled? ☐ Can internal deployment credentials be exposed?

What should an assessment deliver?

- A clearly defined and authorised test scope.
- Documented test scenarios.
- Evidence of identified weaknesses.
- An explanation of technical and business impact.
- Prioritised remediation recommendations.
- Retesting where included in the engagement.

SCOPE AND LIMITS

Not every model extraction risk can be confirmed through a short assessment. The feasibility of extraction depends on the model, its interface, the information returned and the resources available to an attacker.

The objective is to establish what can realistically be exposed, under which conditions, and how to reduce that exposure.

07 · THE LINK TECHREATIONS INTELLIGENCE PERSPECTIVE

AI security must protect the intelligence behind the product.

As organisations turn machine learning models into commercial services, they expose more than a user interface. They expose a capability that may represent years of research, engineering and investment.

That capability deserves security controls.

Model stealing demonstrates why AI security cannot stop at protecting servers, databases or model files. The interaction between a model and its users can itself become an attack surface.

Organisations developing or deploying proprietary AI should ask:

Q1	What information does our model reveal through its outputs?
Q2	Can someone systematically query our service to approximate its behaviour?
Q3	Do we detect unusual patterns of model access?
Q4	Are our model assets protected throughout their lifecycle?
Q5	Have we tested these protections under realistic conditions?

OUR POSITION

At LINK TECHREATIONS, we believe AI security must be evaluated at the level of the system itself: its models, APIs, integrations, access controls and operational behaviour.

Our work focuses on identifying weaknesses in AI-enabled applications and systems through controlled technical assessments, AI red teaming and security testing.

LINK TECHREATIONS INTELLIGENCE is our research and analysis initiative, created to make complex AI security threats understandable, examine their technical implications and help organisations make more informed security decisions.

We don't sell fear. We test systems.

The question is no longer only whether your AI works. *It is whether you can protect the intelligence that makes it valuable.*

REFERENCES & FURTHER READING

1. Tramèr et al. (2016), *Stealing Machine Learning Models via Prediction APIs*, USENIX Security Symposium. https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper.pdf

2. Oliynyk, Mayer & Rauber (2023), *I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences*, ACM Computing Surveys. https://doi.org/10.1145/3595292

3. Zhao et al. (2025), *A Survey on Model Extraction Attacks and Defenses for Large Language Models*, ACM SIGKDD 2025. https://doi.org/10.1145/3711896.3736573

4. OWASP, *AI Exchange: Model Exfiltration*. https://owaspai.org/go/modelexfiltration

5. Microsoft, *Model Theft (IP Theft)*. https://learn.microsoft.com/en-us/security/zero-trust/catalog-ai-attack-techniques/model-theft