Survey of Attacks and Defenses on Edge-Deployed Neural Networks

Mihailo Isakov[1], Dr. Vijay Gadepally[2],
Dr. Karen Gettings[2], Dr. Michel A. Kinsy[1]

[1] Boston University, Boston, MA

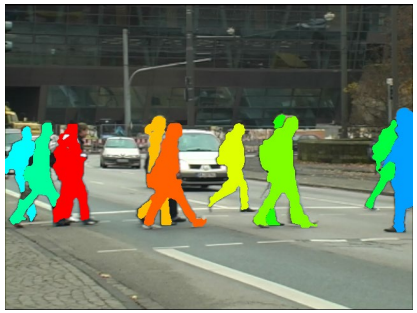
[2] MIT Lincoln Laboratory, Lexington, MA

BOSTON UNIVERSITY

Department of Electrical & Computer Engineering

Neural Network Applications

- Several areas where Deep Neural Networks (DNN) provide state-of-the-art results:
  - Computer Vision
  - Natural Language Processing
  - Robotics



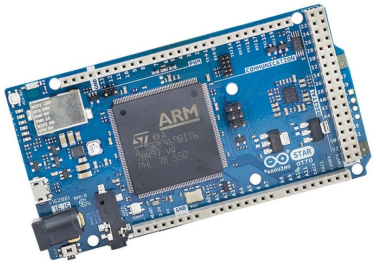
Networks Are Moving To The Edge

- Many users cannot use cloud services

Privacy

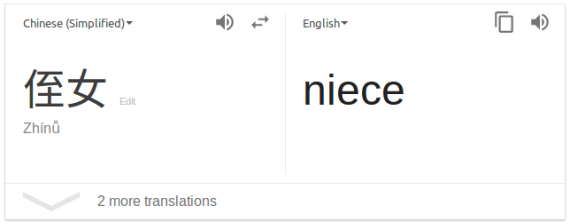
Latency

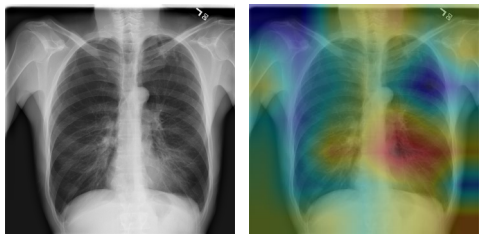
Network Access



Power

DNN Security and Deployment

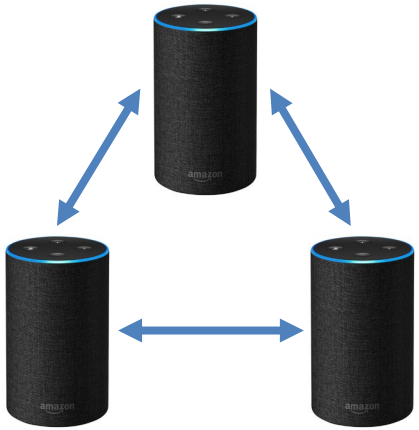




Conventional cybersecurity

Security in the IoT age

Here be dragons



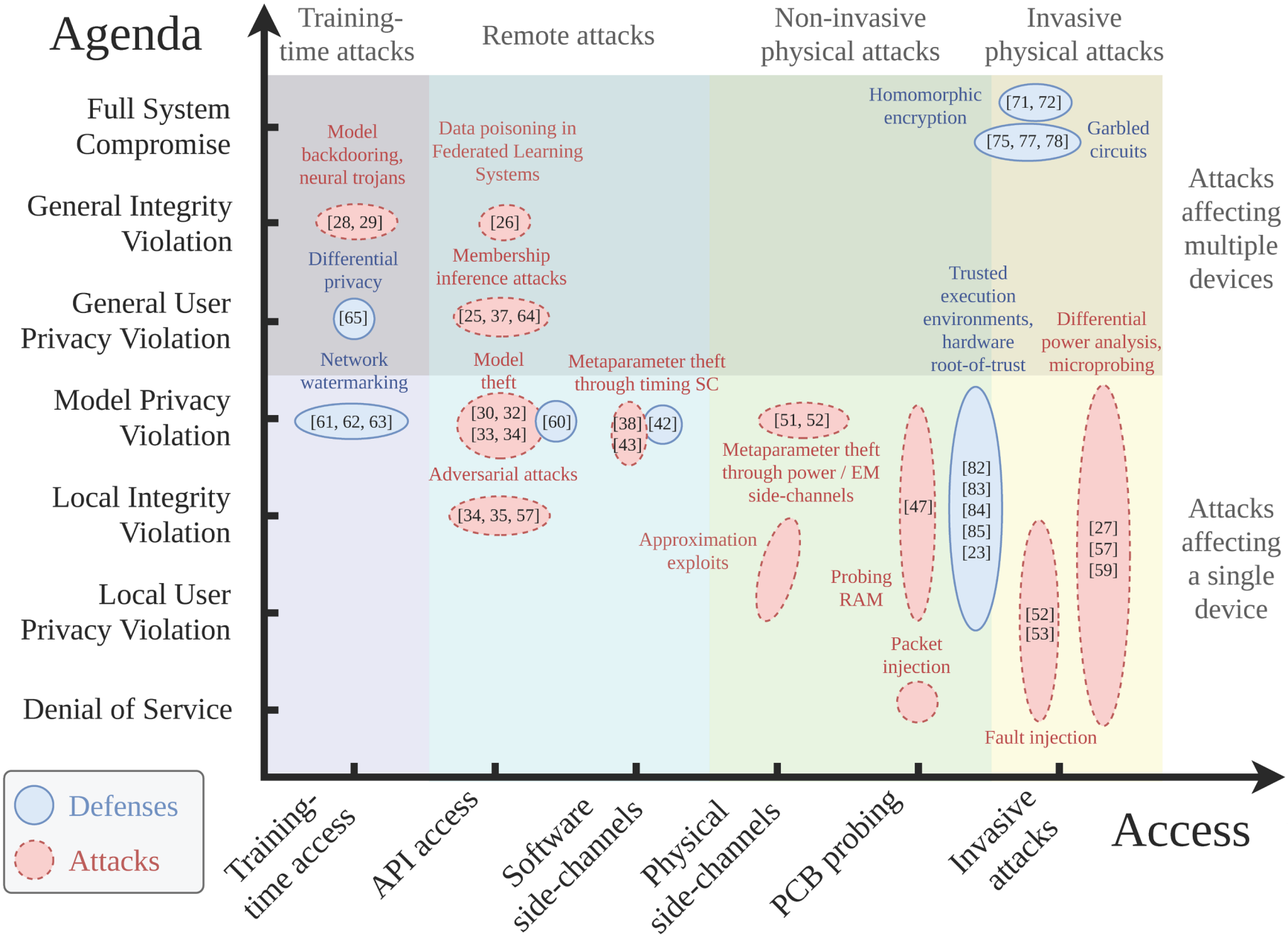
What This Talk Is **Not** About

What This Talk **Is** About

How does the attack surface of edge devices change once they are running DNNs?

Taxonomy of Attacks & Defenses on Edge DNNs

- We try to classify attacks/defenses on two axes:
  - Attacker's level of access to edge device
    - API access
    - Software / hardware side-channels
    - PCB probing
    - Invasive attacks
  - Attacker's agenda
    - Availability violation / DoS
    - User data / DNN model privacy violation
    - Integrity violation


Agenda
Full System Compromise
General Integrity Violation
General User Privacy Violation
Model Privacy Violation
Local Integrity Violation
Local User Privacy Violation
Denial of Service
Training-time attacks
Remote attacks
Non-invasive physical attacks
Invasive physical attacks
Model backdooring, neural trojans
[28, 29]
Differential privacy
[65]
Network watermarking
[61, 62, 63]
Data poisoning in Federated Learning Systems
[26]
Membership inference attacks
[25, 37, 64]
Model theft
[30, 32] [33, 34]
[60]
Adversarial attacks
[34, 35, 57]
Metaparameter theft through timing SC
[38] [43]
[42]
Metaparameter theft through power / EM side-channels
[51, 52]
Approximation exploits
Probing RAM
[47]
Homomorphic encryption
[71, 72]
[75, 77, 78]
Garbled circuits
Trusted execution environments, hardware root-of-trust
[82] [83] [84] [85] [23]
Packet injection
Differential power analysis, microprobing
[27] [57] [59]
[52] [53]
Fault injection
Attacks affecting multiple devices
Attacks affecting a single device
Defenses
Attacks
Training-time access
API access
Software side-channels
Physical side-channels
PCB probing
Invasive attacks
Access

Attacker Agenda: Availability Violation

This sentence is false! →

• • • ←

Attacker Agenda: Availability Violation

This sentence is false! →

Um, true.

I'll go with true. ←

Attacker Agenda: Availability Violation

- DNNs are typically constant-time functions
  - Some data-dependent inference accelerators like Cnvlutin [1] can speed up inference by ignoring zero values in activations
  - Denial of Service does not work on DNNs & CNNs

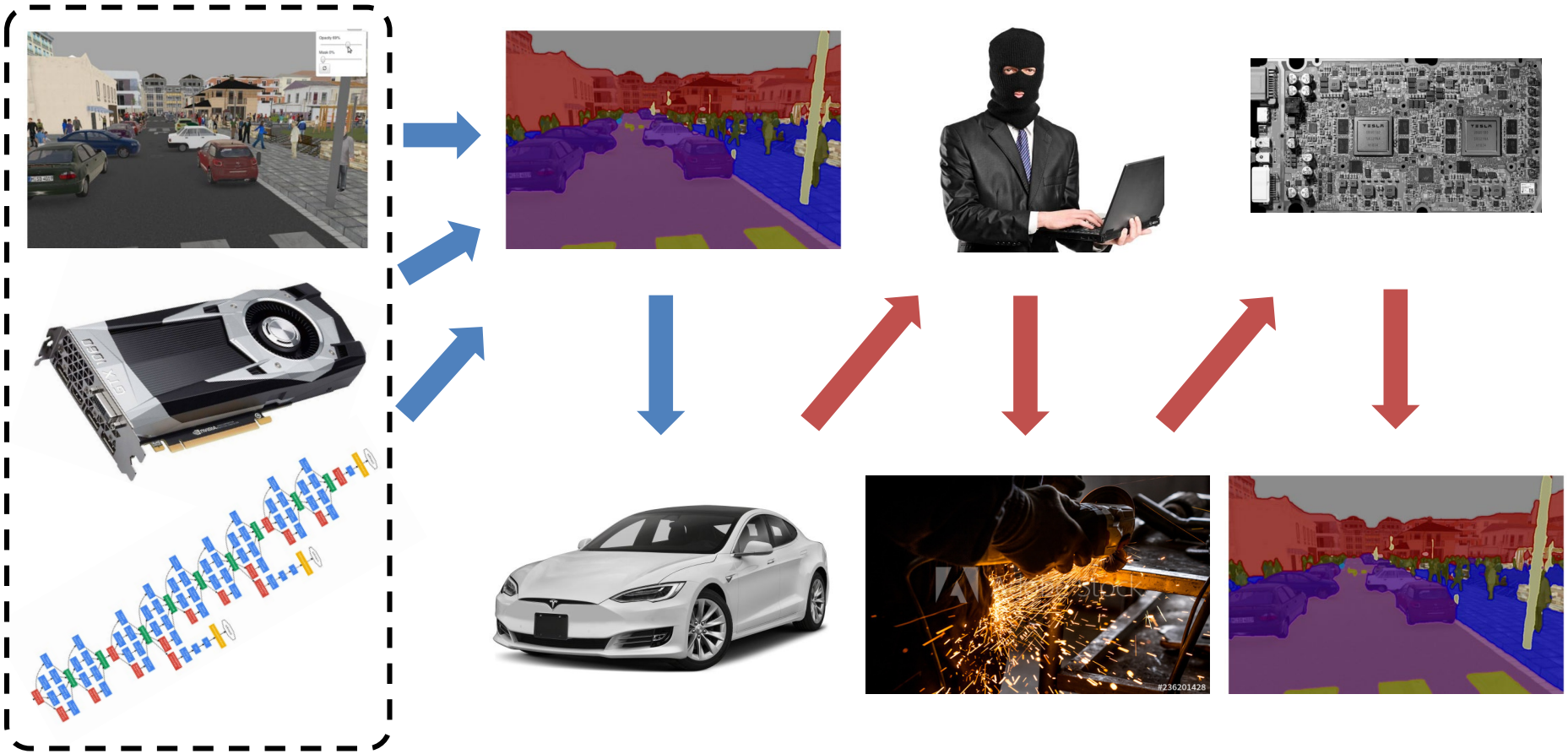
- **Recurrent neural networks may be vulnerable to DoS**

[1] Cnvlutin: Ineffectual-Neuron-Free Deep Neural Network Computing

Attacker Agenda: User Privacy Violation

- Users require private inference on edge devices because:
  - They care about their privacy and don't trust that their data is safe in the cloud
  - There are legal restrictions on moving data to the cloud and data must be processed in-situ

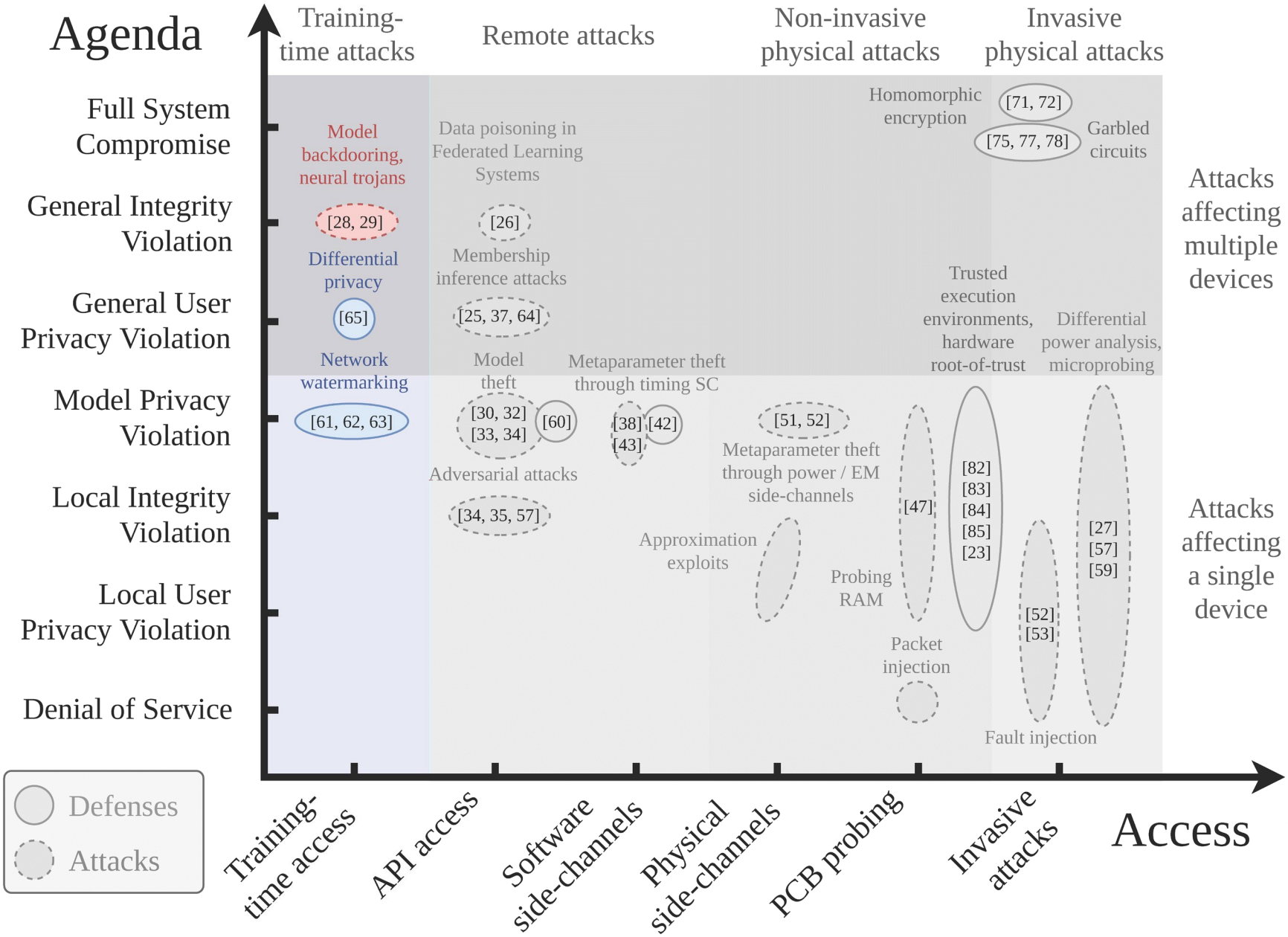
Attacker Agenda: Model Privacy Violation



Years of collecting data, millions of GPU-hours, hundreds of PhDs over several years

Attacker Agenda: Integrity Violation

- Attackers may not want to outright prevent the device from functioning
  - Instead, they may want to force the on-device DNN to perform in an unacceptable way

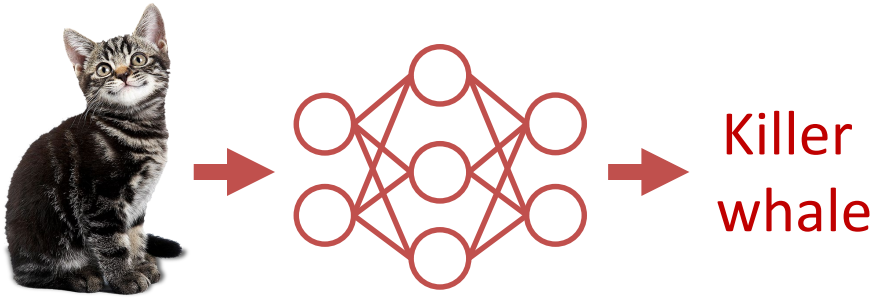

Agenda
Full System Compromise
General Integrity Violation
General User Privacy Violation
Model Privacy Violation
Local Integrity Violation
Local User Privacy Violation
Denial of Service
Training-time attacks
Remote attacks
Non-invasive physical attacks
Invasive physical attacks
Model backdooring, neural trojans
[28, 29]
Differential privacy
[65]
Network watermarking
[61, 62, 63]
Data poisoning in Federated Learning Systems
[26]
Membership inference attacks
[25, 37, 64]
Model theft
[30, 32] [33, 34]
[60]
Adversarial attacks
[34, 35, 57]
Metaparameter theft through timing SC
[38] [43]
[42]
[51, 52]
Metaparameter theft through power / EM side-channels
Approximation exploits
Homomorphic encryption
[71, 72]
[75, 77, 78]
Garbled circuits
Trusted execution environments, hardware root-of-trust
Differential power analysis, microprobing
[47]
Probing RAM
[82] [83] [84] [85] [23]
Packet injection
[52] [53]
[27] [57] [59]
Fault injection
Attacks affecting multiple devices
Attacks affecting a single device
Defenses
Attacks
Training-time access
API access
Software side-channels
Physical side-channels
PCB probing
Invasive attacks
Access

Training Time Attacks / Defenses

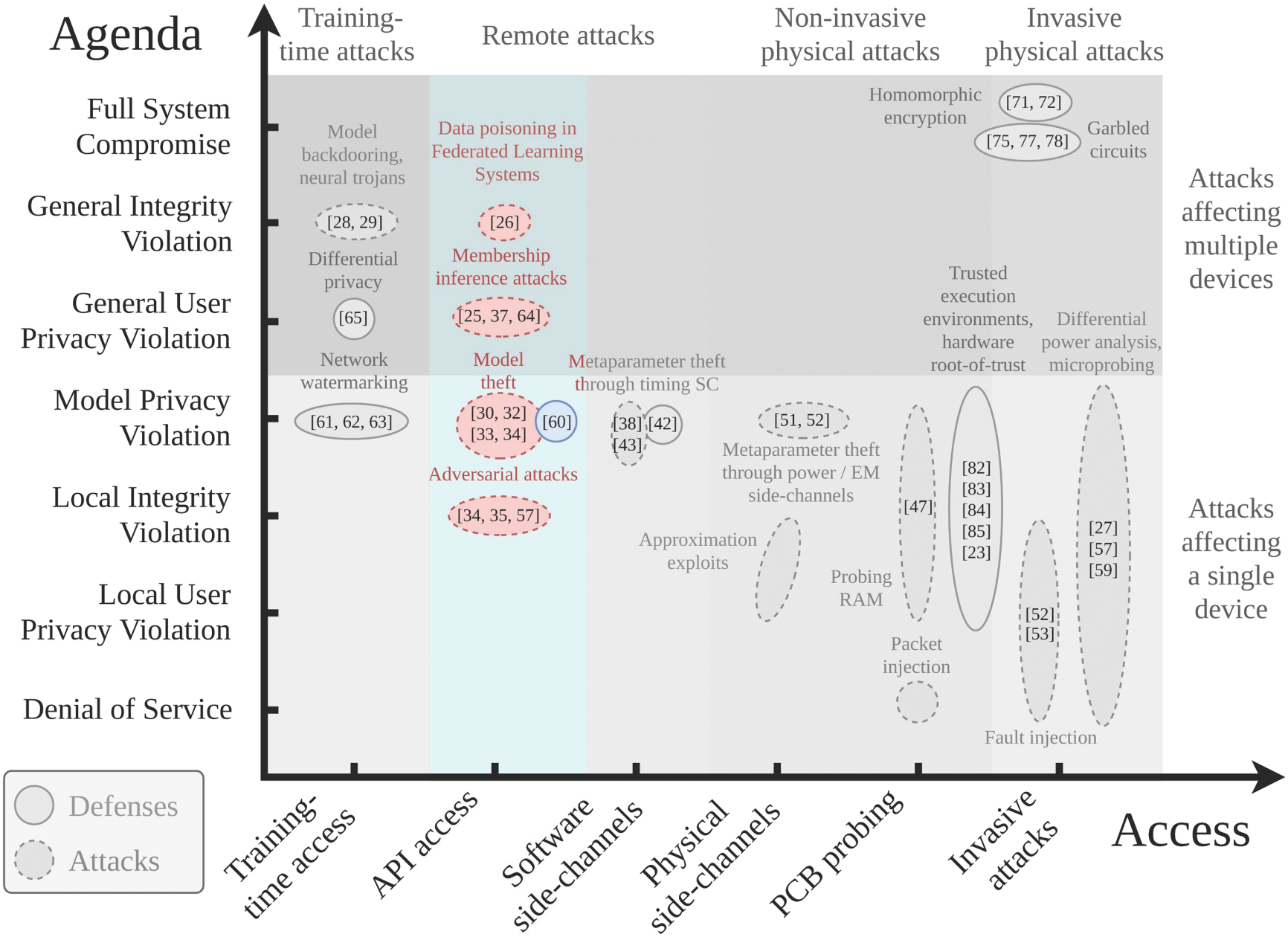
- Supply chain attacks
  - Neural Trojans
  - Backdoored DNNs

[1]

- Model Privacy defenses:
  - DNN watermarking

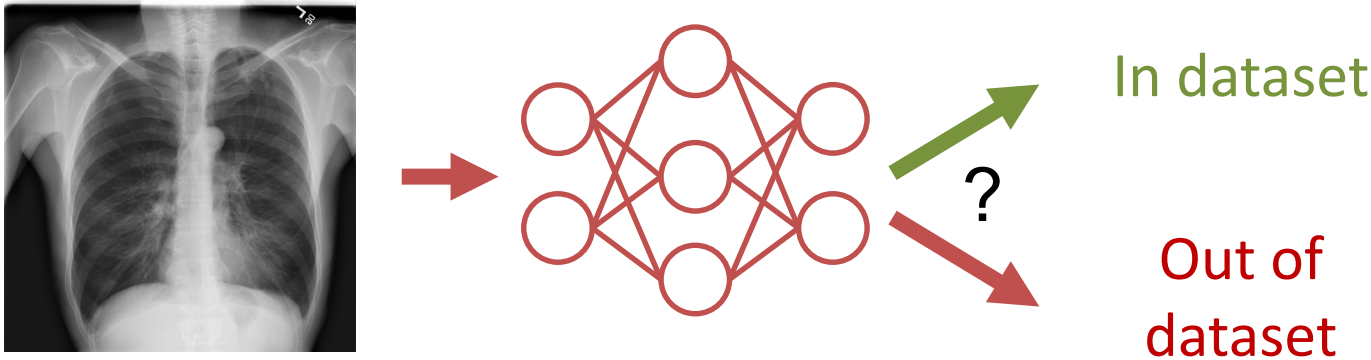


[1] BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain


Agenda
Full System Compromise
General Integrity Violation
General User Privacy Violation
Model Privacy Violation
Local Integrity Violation
Local User Privacy Violation
Denial of Service
Training-time attacks
Remote attacks
Non-invasive physical attacks
Invasive physical attacks
Model backdooring, neural trojans
[28, 29]
Differential privacy
[65]
Network watermarking
[61, 62, 63]
Data poisoning in Federated Learning Systems
[26]
Membership inference attacks
[25, 37, 64]
Model theft
[30, 32] [33, 34]
[60]
Adversarial attacks
[34, 35, 57]
Metaparameter theft through timing SC
[38] [43]
[42]
Metaparameter theft through power / EM side-channels
[51, 52]
Approximation exploits
Probing RAM
[47]
Packet injection
Homomorphic encryption
[71, 72]
[75, 77, 78]
Garbled circuits
Trusted execution environments, hardware root-of-trust
[82] [83] [84] [85] [23]
Differential power analysis, microprobing
[27] [57] [59]
[52] [53]
Fault injection
Attacks affecting multiple devices
Attacks affecting a single device
Defenses
Attacks
Training-time access
API access
Software side-channels
Physical side-channels
PCB probing
Invasive attacks
Access

API-Level Attacks

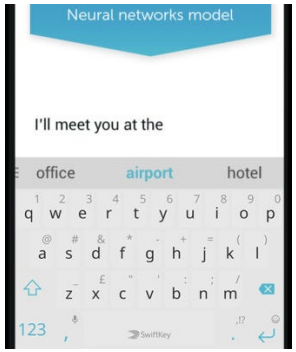
- API-level attacks:
  - Adversarial Examples
  - Membership Inference
  - Model Theft

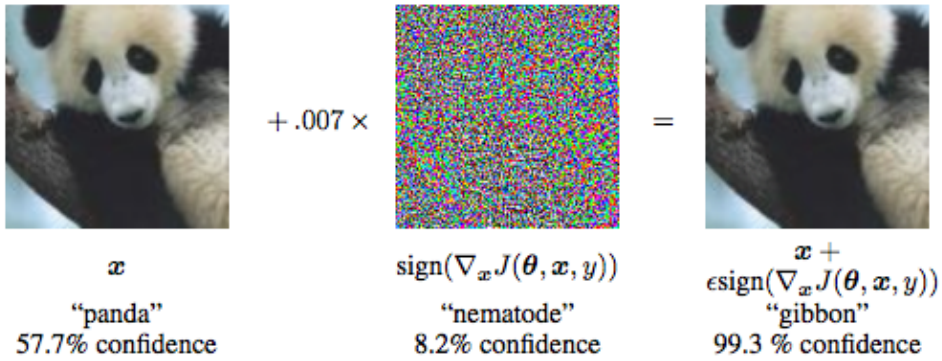


N. Carlini, The secret sharer: Measuring unintended neural network memorization & extracting secrets

Model Theft

- Stealing ML models gives the attackers:
  - A way to bypass paying for the service
  - An expensive model they can sell
  - A way to peek intro the training data
  - A faster way of generating adversarial examples
  - A way of evading detection from ML models?

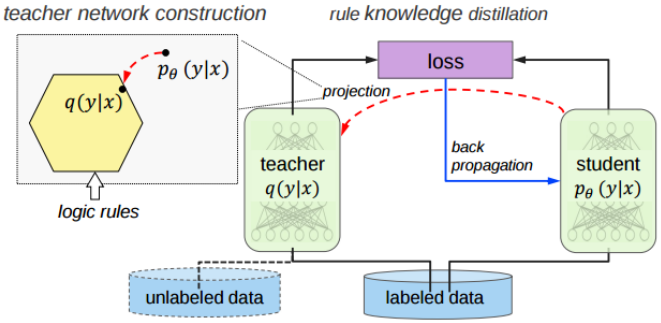




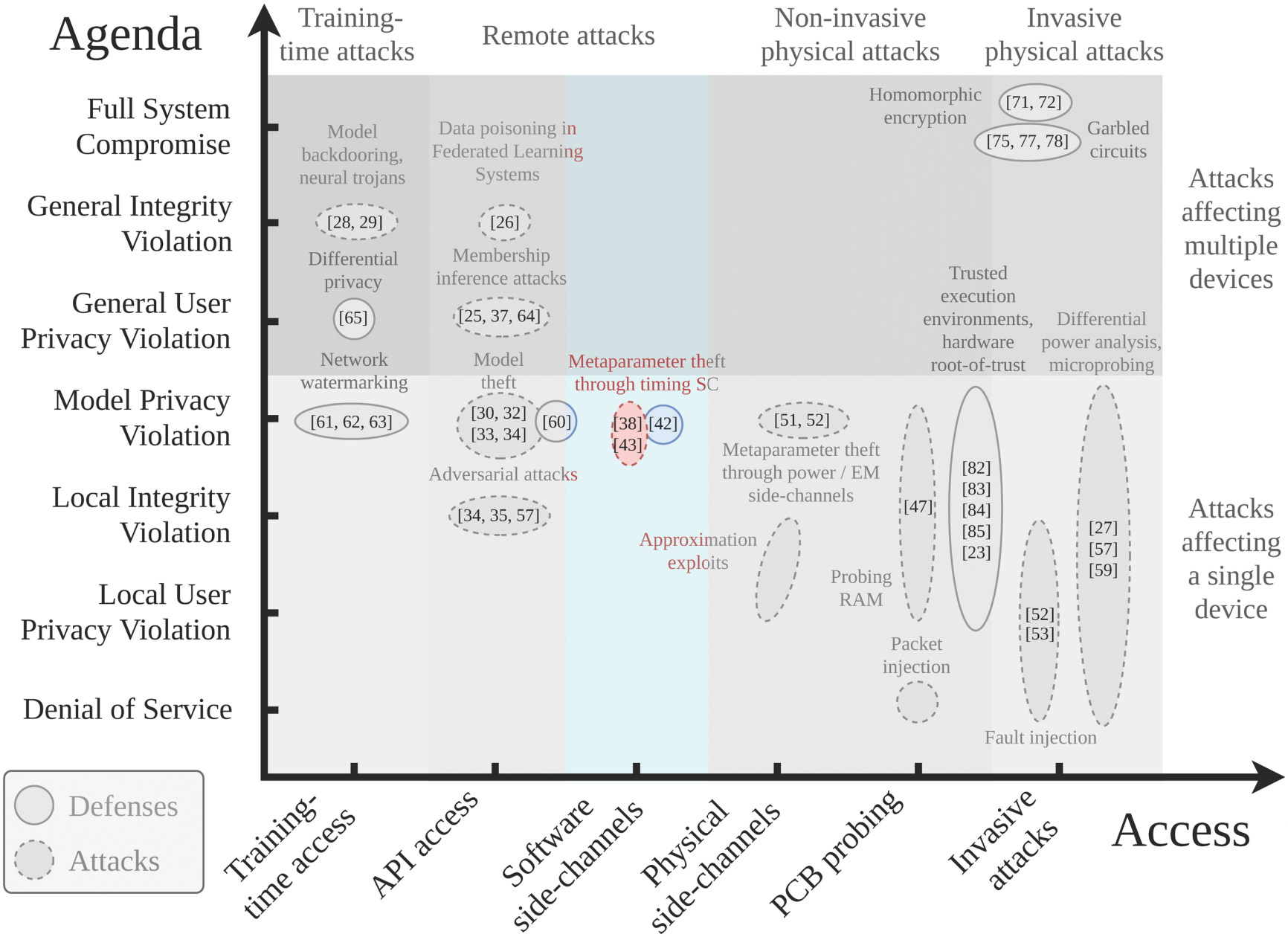
Model Theft: Crafted-Input Attacks

- Models are just big equations, where each parameter is an unknown variable
- By collecting input-output pairs, the attacker can reconstruct the model
  - A lot cheaper than training your own, **and you don't need the dataset!**

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$



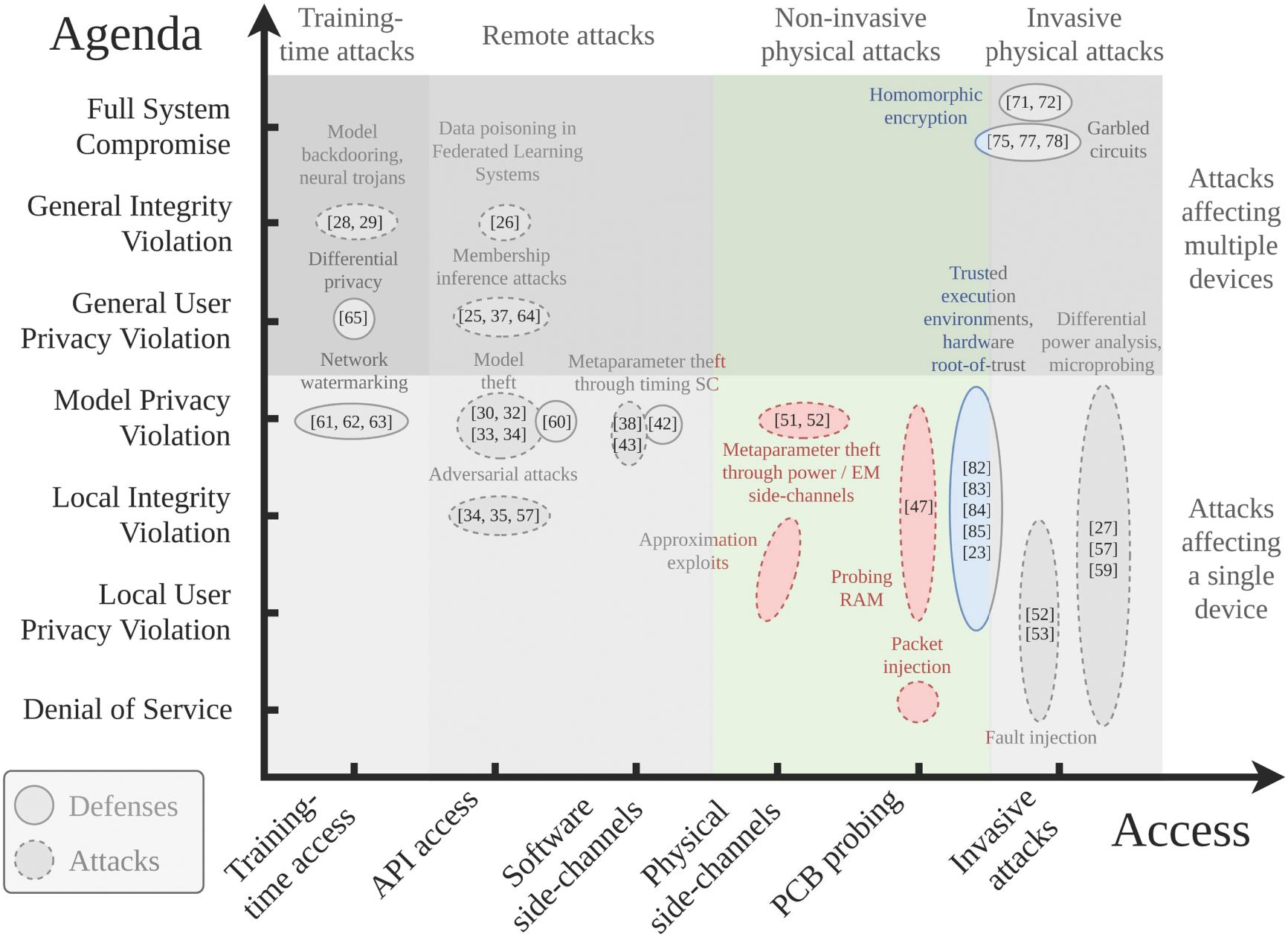
F. Tramer, Stealing machine learning models via prediction APIs
S. J. Oh, Towards Reverse-Engineering Black-Box Neural Networks


Agenda
Full System Compromise
General Integrity Violation
General User Privacy Violation
Model Privacy Violation
Local Integrity Violation
Local User Privacy Violation
Denial of Service
Training-time attacks
Remote attacks
Non-invasive physical attacks
Invasive physical attacks
Model backdooring, neural trojans
[28, 29]
Differential privacy
[65]
Network watermarking
[61, 62, 63]
Data poisoning in Federated Learning Systems
[26]
Membership inference attacks
[25, 37, 64]
Model theft
[30, 32] [33, 34]
[60]
Adversarial attacks
[34, 35, 57]
Metaparameter theft through timing SC
[38] [43]
[42]
[51, 52]
Metaparameter theft through power / EM side-channels
Approximation exploits
Homomorphic encryption
[71, 72]
[75, 77, 78]
Garbled circuits
Trusted execution environments, hardware root-of-trust
Differential power analysis, microprobing
[47]
Probing RAM
[82] [83] [84] [85] [23]
Packet injection
[52] [53]
[27] [57] [59]
Fault injection
Attacks affecting multiple devices
Attacks affecting a single device
Defenses
Attacks
Training-time access
API access
Software side-channels
Physical side-channels
PCB probing
Invasive attacks
Access

Software Side-Channel Attacks

- Majority of software side-channel attacks target cache
  - Memory access patterns can reveal neural network architecture
    - Number of layers, size of layers, type of each layer
  - If the attacker knows the framework used for developing the DNN, they can use instruction cache side-channels to monitor function invocation

M. Yan, Cache Telepathy: Leveraging Shared Resource Attacks to Learn DNN Architectures
V. Duddu, Stealing neural networks via timing side channels
S. Hong, Security analysis of deep neural networks operating in the presence of cache sidechannel attacks


Agenda
Full System Compromise
General Integrity Violation
General User Privacy Violation
Model Privacy Violation
Local Integrity Violation
Local User Privacy Violation
Denial of Service
Training-time attacks
Remote attacks
Non-invasive physical attacks
Invasive physical attacks
Model backdooring, neural trojans
[28, 29]
Differential privacy
[65]
Network watermarking
[61, 62, 63]
Data poisoning in Federated Learning Systems
[26]
Membership inference attacks
[25, 37, 64]
Model theft
[30, 32] [33, 34]
[60]
Adversarial attacks
[34, 35, 57]
Metaparameter theft through timing SC
[38] [43]
[42]
[51, 52]
Metaparameter theft through power / EM side-channels
Approximation exploits
Probing RAM
[47]
Packet injection
Homomorphic encryption
[71, 72]
[75, 77, 78]
Garbled circuits
Trusted execution environments, hardware root-of-trust
[82] [83] [84] [85] [23]
Differential power analysis, microprobing
[27] [57] [59]
[52] [53]
Fault injection
Attacks affecting multiple devices
Attacks affecting a single device
Defenses
Attacks
Training-time access
API access
Software side-channels
Physical side-channels
PCB probing
Invasive attacks
Access

Non-Invasive Physical Attacks

- Attacks that require physical access to the edge device, but do not require destructive measures
  - Physical side-channels:
    - Power, electro-magnetic, etc.
    - In [1] authors recover input image by observing the power usage of the first convolutional filter
  - Probing DRAM can reveal the DNN weights and DNN architecture
    - Even if the DNN is run in a secure enclave, DRAM access patterns can reveal network architecture [2]

[1] L. Wei, I know what you see: Power side-channel attack on convolutional neural network accelerators
[2] W. Hua, Reverse Engineering Convolutional Neural Networks Through Side-channel Information Leaks

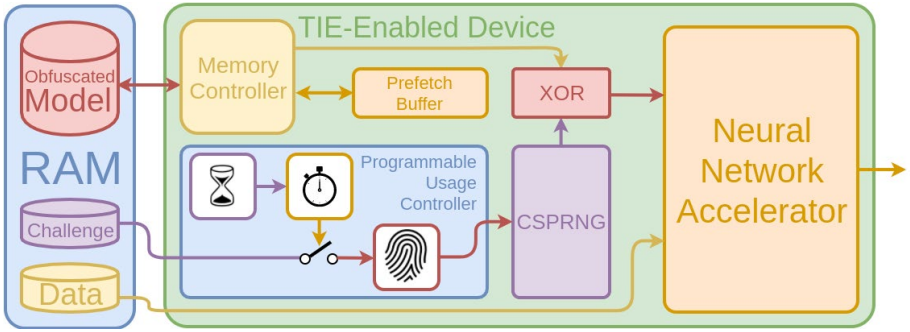
Secure Deployment of DNNs

- Two directions for secure deployment of DNNs:
  - Secure Enclave-based defenses
  - Secure DNN Accelerators
- Enclave-based defenses:
  - MLCapsule [1]: a formally verified SGX-based private DNN inference
  - Slalom [2]: uses SGX to process nonlinearities, and offloads linear operations to insecure CPU/GPU
    - Encrypts results sent to insecure hardware, and attests results
    - Applied nonlinear functions within the enclave

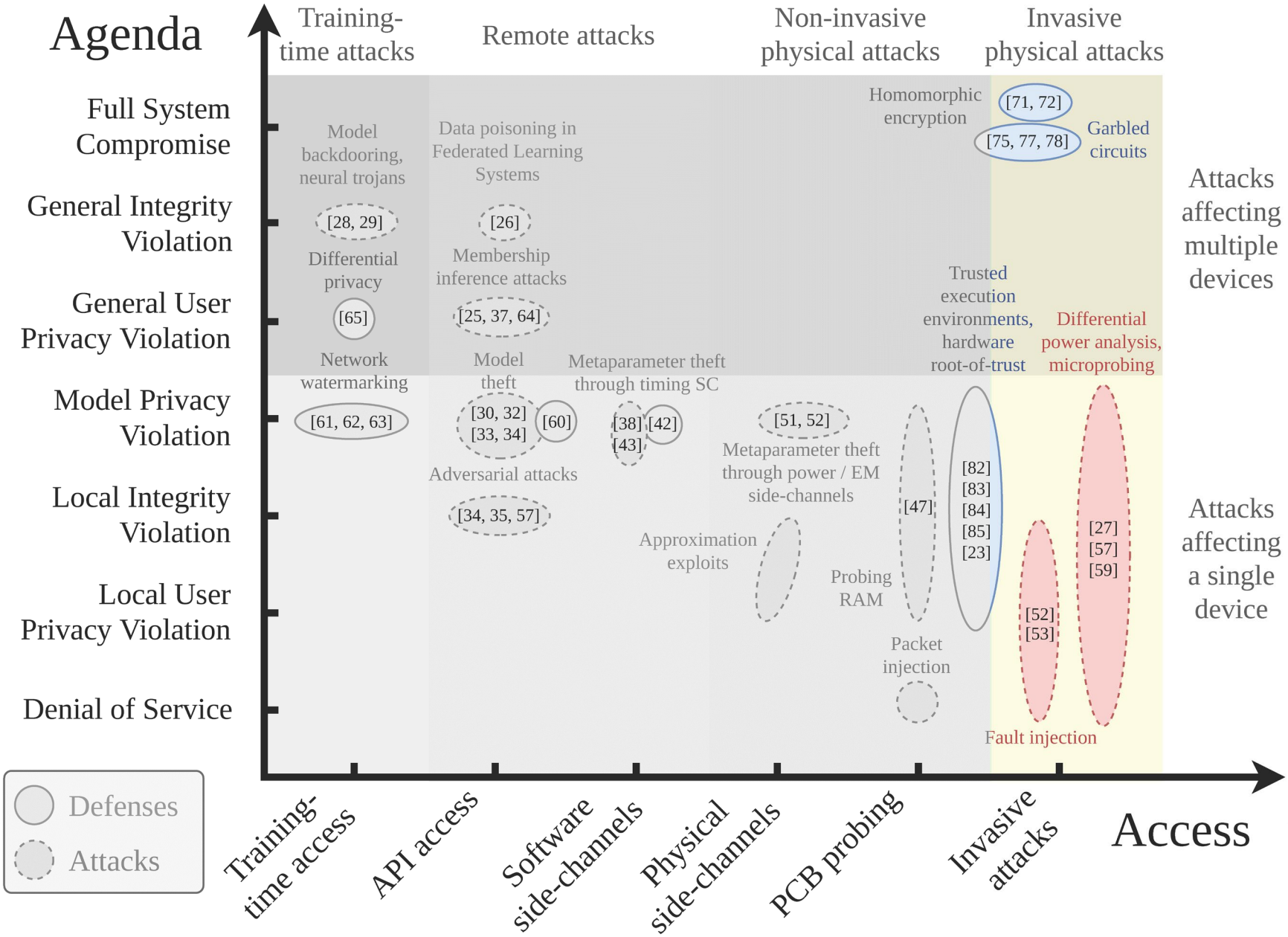
[1] L. Hanzlik, MLCapsule: Guarded Offline Deployment of Machine Learning as a Service
[2] F. Tramer, Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware

Secure DNN Accelerators

- Trusted Inference Engine [1]:
  - Prevents attackers from stealing deployed DNNs
  - Uses a CSPRNG seeded with a PUF to decrypt the stored model
  - Has built-in Programmable Usage Controller (PUC) to prevent attackers from extracting model through API
  - Buffers DRAM accesses to prevent attackers from learning DNN architecture



[1] M. Isakov, Preventing neural network model exfiltration in machine learning hardware accelerators


Agenda
Full System Compromise
General Integrity Violation
General User Privacy Violation
Model Privacy Violation
Local Integrity Violation
Local User Privacy Violation
Denial of Service
Training-time attacks
Remote attacks
Non-invasive physical attacks
Invasive physical attacks
Model backdooring, neural trojans
[28, 29]
Differential privacy
[65]
Network watermarking
[61, 62, 63]
Data poisoning in Federated Learning Systems
[26]
Membership inference attacks
[25, 37, 64]
Model theft
[30, 32]
[33, 34]
[60]
Adversarial attacks
[34, 35, 57]
Metaparameter theft through timing SC
[38]
[43]
[42]
[51, 52]
Metaparameter theft through power / EM side-channels
Approximation exploits
Homomorphic encryption
[71, 72]
[75, 77, 78]
Garbled circuits
Trusted execution environments, hardware root-of-trust
Differential power analysis, microprobing
[47]
Probing RAM
[82]
[83]
[84]
[85]
[23]
Packet injection
[52]
[53]
[27]
[57]
[59]
Fault injection
Attacks affecting multiple devices
Attacks affecting a single device
Defenses
Attacks
Training-time access
API access
Software side-channels
Physical side-channels
PCB probing
Invasive attacks
Access

Invasive Physical Attacks

- Assume that attackers are able to access the internals of all chips on the device
- Include:
  - Decapsulation
  - Microprobing
  - Fault Injection
  - Cold boot attacks on DRAM
- With this level of attacker access, we cannot count on Hardware Root-of-Trust

J. Breier, DeepLaser: Practical Fault Attack on Deep Neural Networks
Y. Liu, Fault Injection Attack on Deep Neural Network
A. S. Rakin, Bit-Flip Attack: Crushing Neural Network with Progressive Bit Search

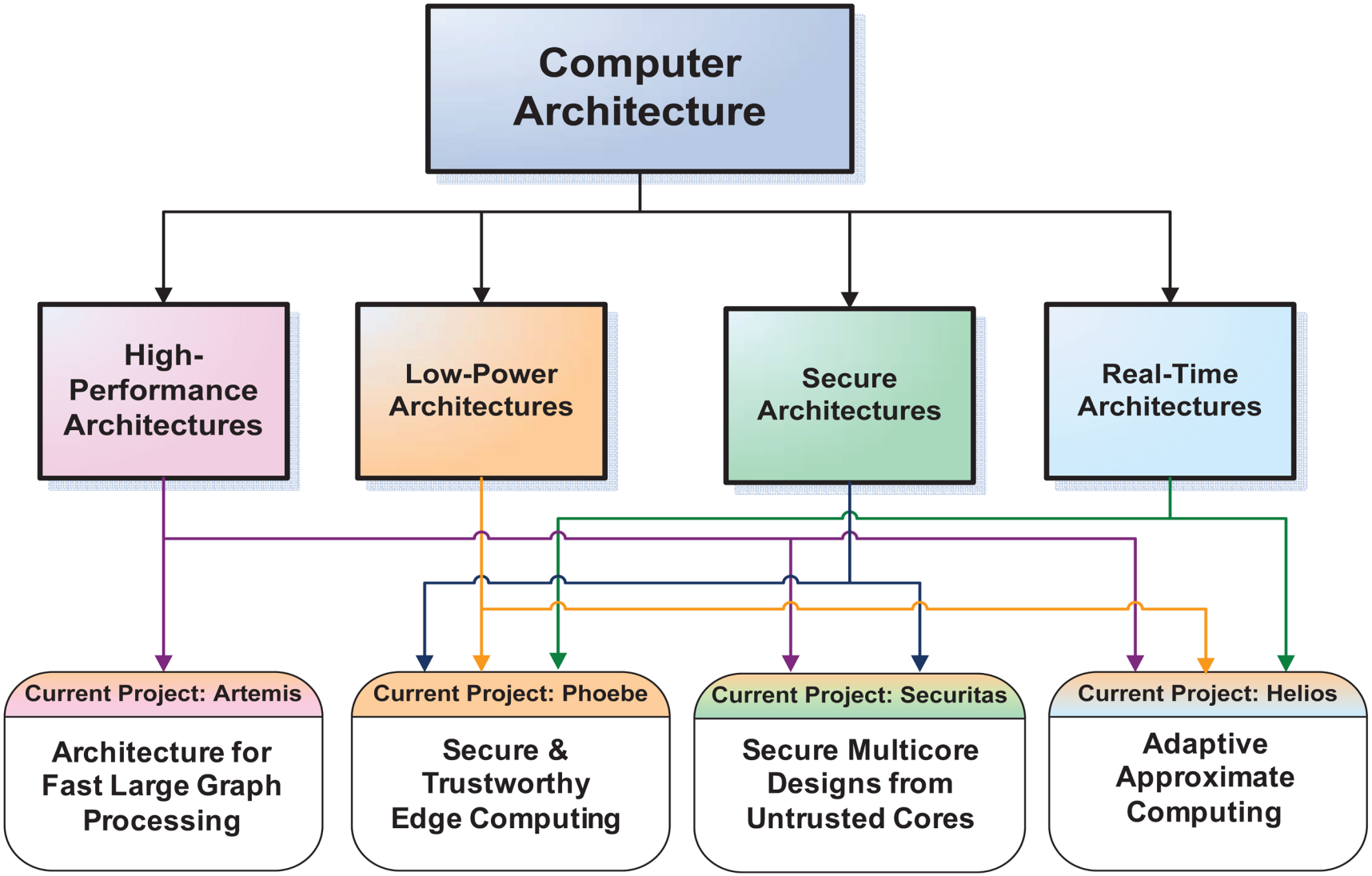
Defenses Against Invasive Attacks

- Homomorphic Encryption:
  - CryptoNets [1]: Runs DNNs on encrypted inputs
    - 100-1000x slower than computing on plain inputs
  - If performed in the cloud, requires communication
  - If performed on-device, requires significant amounts of power and time
- Yao's Garbled Circuits:
  - Allow performing private DNN inference with one party supplying the model, and other the inputs
    - Can be accelerated using pruning [2] or by replacing multiplications with XNOR operations [3]
  - Require multiple rounds of communication with the cloud

[1] N. Dowlin, CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy
[2] B. Darvish, DeepSecure: Scalable Provably-Secure Deep Learning
[3] M. S. Riazi, XONN: xnor-based oblivious deep neural network inference

Conclusion

- DoS attacks on DNNs are an under-explored area
- Hardware Root-of-Trust solves privacy and integrity issues unless targeted by a sophisticated attacker
- Homomorphic Encryption & Garbled Circuits may be too expensive for edge devices
- Algorithmic attacks such as API-based model theft and adversarial examples are still an open problem and need an algorithmic solution


Computer Architecture
High-Performance Architectures
Low-Power Architectures
Secure Architectures
Real-Time Architectures
Current Project: Artemis
Architecture for Fast Large Graph Processing
Current Project: Phoebe
Secure & Trustworthy Edge Computing
Current Project: Securitas
Secure Multicore Designs from Untrusted Cores
Current Project: Helios
Adaptive Approximate Computing