Solution Brief

Enterprise Network Security
Solution Provider, Independent Software Vendor

Lanner Delivers AI Security Appliance for Network Edge

Lanner NCA-4240, based on 14th Gen Intel® Core™ desktop processors, is optimized with performance for AI-based next-generation firewall (NGFW) and cost, power consumption, and form factor for branch office applications

In the last 10 years, firewalls have evolved and added new features to provide better defense against fast changing cybersecurity threats.

Firewalls were originally designed to examine inbound packet payloads against a static database of security policies and rules to allow or deny access to the enterprise network based on those rules. With improvements in virtualization software and CPU performance, firewalls can do this packet inspection at wire speed on Intel® architecture CPUs.

Firewalls are placed at network ingress points and thus are the first line of defense. Because of the importance of the firewall, technology has evolved to add new security services that can leverage packet lookup and database capabilities. These next generation firewalls (NGFW) combine firewall capabilities with other security services including antivirus, intrusion protection system, application identification and others.

Now NGFWs are evolving again to stay ahead of a new class of cybersecurity threats such as ransomware, zero-day attacks, phishing attacks as well as to do a better job against ongoing threats such as distributed denial of service (DDoS) attacks. This firewall evolution uses artificial intelligence / machine learning (AL/ML) to make the rules database more dynamic. New AI-NGFWs now feature AI-driven detection capabilities that can help it recognize and react to cyber threats more quickly.

With this capability, the AI-NGFW can be trained on a massive database of known threats that allows it to infer new threat variants and respond automatically to stop a threat. The use of AI also allows the NGFW to efficiently respond to threats.

NGFWs have long been deployed in branch offices and other locations at the edge of the network in order to harden these potential paths for a cyber-attack. But adding AI to these systems means increasing the compute performance required to run these applications.

Lanner Electronics, an Intel® Network Builders ecosystem member, has developed the NCA-4240, a network security appliance with the features and performance to enable AI-powered NGFWs running in branch offices or network edge locations to fortify these network locations with a system that can proactively monitor and adapt to evolving cyber threats in real-time.

Lanner NCA-4240 Delivers Branch Office AI-NGFWs

The Lanner NCA-4240 is a 1U rackmount appliance (see Figure 1) that is designed for branch office applications with power-efficient performance and extensive networking. The server features a built-in 1 GbE RJ45 port via Intel® Ethernet Controller I219 and eight 2.5 GbE RJ45 ports via an embedded Intel® Ethernet Controller I226. Additional connectivity is available via a slot that can support a discrete network interface card with speeds up to 100GbE.

Figure 1. Lanner NCA-4240 front view.

Three pairs of Gen 3 bypass connections are available for redundant out-of-band communications that is a requirement to maintain communications if the networking ports are disabled. With these bypass ports, traffic can continue to be forwarded through the system.

The server has fast memory supporting up to 64GB of DDR5 4800 MHz RAM via two 288-pin DIMM, which is essential to support the compute-heavy AI-NGFW workloads. For storage, the server supports two 2.5" HDD/SSD storage modules. The server can support additional features via PCIe x8 or M.2 slots.

Performance from 14th Gen Intel® Core™ Desktop Processor Family

The Lanner NCA-4240 server is based on 14th Gen Intel Core desktop processor family, which feature a hybrid architecture combining performance-cores (P-cores) and efficient-cores (E-cores), on a single processor die with a maximum of up to 24 cores (eight Performance-cores and 16 Efficient-cores) and up to 32 threads. Models in the CPU family have a very energy efficient 65 Watt thermal design power (TDP).

Integrated GPU for AI Processing

The CPUs offer enhanced AI processing power via an integrated Intel® UHD Graphics 770 GPU. This iGPU functionality uses the Intel® Xe Architecture to boost the performance of AI applications by delivering faster inferencing for network security workloads.

The GPUs feature 32 graphics execution units allowing a high degree of parallelization for AI workloads, combined with built-in AI acceleration from Intel® Deep Learning Boost (Intel® DL Boost) and the Intel® Distribution of OpenVINO™ toolkit. The use of integrated GPU offers significant AI performance to meet branch office requirements and offers a cost-effective solution by bringing the overall total cost of operation (TCO) down as it does not need a discrete GPU card for AI.

The CPU also features PCIe 5.0 ready/PCIe 4.0 slots for additional functionality, USB 3.2 Gen 2x2 ports and support for discrete Wi-Fi 6E.

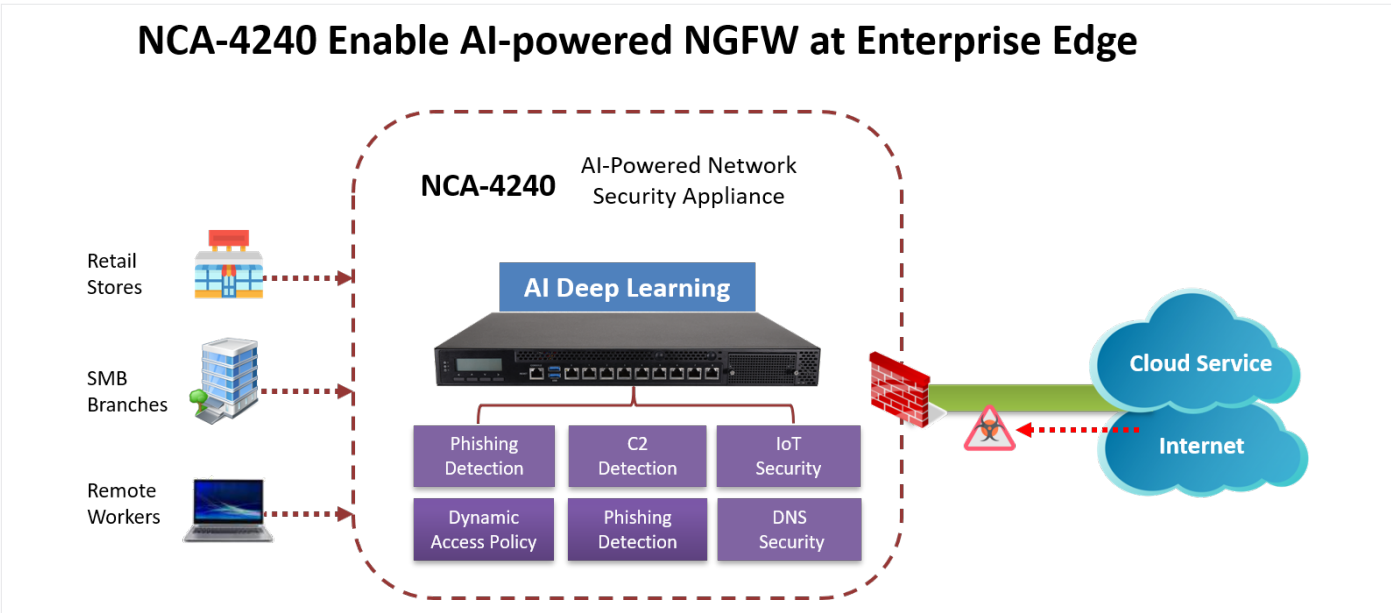

Figure 2. NCA-4240 in a typical branch office AI-NGFW application.

As shown in Figure 2, the Lanner NCA-4240 empowers AI-driven NGFWs at the enterprise edge. In this picture, data traffic from network users (on the left) passes through the NCA-4240 where all seven layers of each packet are examined against the dynamic databases shown in purple. The use of AI means these databases are updated based on AI training. The performance of the server and its low power consumption enables AI-based NGFWs from any third-party ISV to run on the systems and improves the software's capacity to identify, counteract, and alleviate advanced and evolving cyber threats.

Test Set Up

To show the AI inference performance under NCA-4240, Lanner and Intel chose to test[1] some typical AI-NGFW use cases that utilize NGFW from four industry-leading vendors. The use cases are email phishing and malware portable executable (PE). The tests measured latency using three different AI frameworks: TensorFlow, ITEX and Open VINO. The 14th Gen Intel® Core™ desktop processor inside the Lanner NCA-4240 was configured in a number of ways to examine the performance using the iGPU and the P-core and E-core. The iGPU offered significantly lower latency figures at both FP32 and INT8 then either of the two CPU cores tested.

For the PE tests, the open-source MalConv model was used (https://github.com/elastic/ember/tree/master/malconv). First, the H5 model was converted to the FP32 model and then used Intel® Neural Compressor to quantize it to the INT8 model. Detail codes and methods can be found in this downloadable PDF. We benchmarked the NCA-4240 using one E-core and one P-core.

From the results, the performance can be boost up to 3.03 x and the inference time can be less than 11.48 ms by quantized model to INT8 by using Intel Neural Compressor.

Framework	Platform		Latency (ms)		Performance Improvement INT8 vs FP32 W/oneDNN
			FP32 w/oneDNN	INT8	
TensorFlow (2.14.0)	Lanner NCA-4240 (Turbo on)	1 P-core	34.73	11.48	**3.03 X**
		1 E-core	80.77	42.85	**1.88 X**

The tests used ITEX to offload the AI workload to iGPU to further improve the AI inference performance. Here are the steps (details can be found at https://github.com/intel/intel-extension-for-tensorflow):

Step 1: Install iGPU driver, details steps can be found at (https://github.com/intel/compute-runtime)

Step 2: Install Intel® oneAPI Base Toolkit 2024.0.0

```
# wget https://registrationcenter-download.intel.com/akdlm//IRC_NAS/20f4e6a1-6b0b-4752-b8c1-e5eacba10e01/l_BaseKit_p_2024.0.0.49564.sh

# sh l_BaseKit_p_2024.0.0.49564.sh

# source /opt/intel/oneapi/compiler/latest/env/vars.sh

# source /opt/intel/oneapi/mkl/latest/env/vars.sh
```

Step 3:

```
# apt install -y clinfo

# clinfo   //Check driver status with clinfo

# apt install -y intel-gpu-tools

# intel_gpu_top  //# Inspect iGPU status and monitor the GPU usage and frequency status.
```

From there it is possible to offload the AI workload from the CPU to the iGPU with zero code changes by only running the following command –

```
# pip install intel-extension-for-tensorflow[xpu]==2.14.0.2
```

Then iGPU can be used to do AI inference.

Framework	Platform		Latency (ms)		Performance Improvement INT8 vs FP32 W/oneDNN
			FP32 w/oneDNN	INT8	
ITEX (2.14.0.2)	Lanner NCA-4240	iGPU	10.47	5.19	**2.02 X**

The AI inference latency can be further improved to 5.19ms. That is 15.56X boost from using 1 E-core to using iGPU and with zero code changes.

A similar test for phishing was also run using the bidirectional encoder representations from transformers (BERT) model. BERT is a neural network model that is trained to detect phishing email from text within the email.

In this test, we used OpenVINO as deep learning framework instead of TensorFlow. The following codes can use OpenVINO NNCF to quantize the model.

```
import os
from pathlib import Path
import datasets
import numpy as np
import nncf
from nncf.parameters import ModelType
import openvino as ov
import torch
from transformers import BertForSequenceClassification, BertTokenizer

MODEL_DIR = "models"
os.makedirs(MODEL_DIR, exist_ok=True)
MAX_SEQ_LENGTH = 512

def load_model(inputs, input_info):
    ir_model_xml = Path(MODEL_DIR) / "bert-base-cased.xml"
    core = ov.Core()
    torch_model = BertForSequenceClassification.from_pretrained('bert-base-cased')
    torch_model.eval

    # Convert the PyTorch model to OpenVINO IR FP32.
    if not ir_model_xml.exists():
        model = ov.convert_model(torch_model, example_input=inputs, input=input_info)
        ov.save_model(model, str(ir_model_xml))
    else:
        model = core.read_model(ir_model_xml)

    return model

def create_data_source():
    raw_dataset = datasets.load_dataset('glue', 'mrpc', split='validation')
    tokenizer = BertTokenizer.from_pretrained('bert-base-cased')

    def _preprocess_fn(examples):
        texts = (examples['sentence1'], examples['sentence2'])
        result = tokenizer(*texts, padding='max_length', max_length=MAX_SEQ_LENGTH, truncation=True)
        result['labels'] = examples['label']
        return result
    processed_dataset = raw_dataset.map(_preprocess_fn, batched=True, batch_size=1)

    return processed_dataset

def nncf_quantize(model, inputs):
    INPUT_NAMES = [key for key in inputs.keys()]
    data_source = create_data_source()

    def transform_fn(data_item):
        inputs = {
            name: np.asarray([data_item[name]], dtype=np.int64) for name in INPUT_NAMES
        }
        return inputs

    calibration_dataset = nncf.Dataset(data_source, transform_fn)
    # Quantize the model. By specifying model_type, we specify additional transformer patterns in the model.
    quantized_model = nncf.quantize(model, calibration_dataset,
                                    model_type=ModelType.TRANSFORMER)

    compressed_model_xml = Path(MODEL_DIR) / "quantized_bert_base_cased.xml"
    ov.save_model(quantized_model, compressed_model_xml)

if __name__ == '__main__':
    input_shape = ov.PartialShape([1, 512])
    input_info = [("input_ids", input_shape, np.int64),("attention_mask", input_shape, np.int64),("token_type_ids", input_shape, np.int64)]
    default_input = torch.ones(1, MAX_SEQ_LENGTH, dtype=torch.int64)
    inputs = {
        "input_ids": default_input,
        "attention_mask": default_input,
        "token_type_ids": default_input,
    }

    model = load_model(inputs, input_info)
quantized_model = nncf_quantize(model, inputs)
```

The BERT base model was benchmarked with 512 max tokens inference time under Lanner NCA-4240 P-core and E-core. Similar to the testing done for the MalConv model, the offloading of the AI workload to the iGPU was done using commands with no new code needed. The command will be similar to `# numactl -C 0 benchmark_app -m models/quantized_bert_base_cased.xml -d GPU -hint latency -shape "[1, 512]"`

The benchmark for the BERT base case model with 512 input tokens:

Framework	Platform		Latency (ms)		Performance Improvement INT8 vs FP32 W/oneDNN
			FP32 model	INT8 model	
OpenVINO (2023.2.0)	Lanner NCA-4240	1 P-core	652.30	253.04	**2.58 X**
		1 E-core	1787.67	956.48	**1.87 X**
		iGPU	119.35	71.14	**1.68 X**

The AI inference latency can be boosted up to 2.58X after using OpenVINO NNCF to quantize the model. The AI inference latency under the iGPU is as low as 71.14ms. That is 25.13X boost from using 1 E-core to using iGPU and without any code changes.

Conclusion

The benchmarks run in this paper make the case for the Lanner NCA-4240 being a good choice for AI NGFW with many options for running AI workloads. The NCA-4240 can benefit from Intel's latest instruction sets and rely on Intel AI ecosystem to boost AI inference performance when running either P-core and E-core. Customers also can save core resources by offloading their AI workloads to iGPU to further improve AI performance without any code changes. The NCA-4240 is a good choice for customers to democratize AI for NGFWs.

Learn More

Network Security Appliance NCA-4240

Intel® Network Builders

14th Gen Intel Core desktop processors

Notices & Disclaimers

[1] The NCA-4240B was used in the testing and is based on the 14th Gen Intel i9-14900 CPU. This processor has Intel DL Boost and the AVX_VNNI instruction sets. This instruction set allows the CPU to work with all the major AI frameworks such as TensorFlow, PyTorch, ONNX and OpenVINO by default. The inference latency can be boosted after the model is quantized to INT8. With Intel® Extension for TensorFlow*, the CPU can offload the AI workload from the CPU to the iGPU which reduces the inference latency.

Performance varies by use, configuration and other factors. Learn more on the Performance Index site.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. No product or component can be absolutely secure.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Your costs and results may vary.

0524/TM/DJA/PDF ♻Please Recycle 359970-001US