



---

# Your NanoSSH™ Evaluation

---

**Version 1.0**

---

**Mocana Corporation**

350 Sansome Street  
Suite 1010  
San Francisco, CA 94104

415-617-0055 Phone  
866-213-1273 Toll Free

info@mocana.com  
www.mocana.com

Copyright © 2008  
Mocana Corp.

---


## Introduction

Thanks for evaluating Mocana NanoSSH™! We want to make sure your evaluation goes smoothly, so if you need any help at all, please don't hesitate to email us at **[support@mocana.com](mailto:support@mocana.com)**. Our expert support team will reply back within one business day. If you have more general questions about device security, or would like strategic advice about how to approach your device project, our support team will be happy to loop in Mocana embedded engineers to discuss your issues and provide suggestions, usually at no charge to you.

This document explains some important technical criteria that might help you evaluate Mocana's NanoSSH product, and its suitability for your particular project. Additionally, these evaluation criteria can help technical decision-makers plan their long-term architectural and device engineering design roadmaps. We invite you to use the following table as a checklist for evaluating your free Mocana NanoSSH trial against competing commercial and open-source offerings.

## My SSH Evaluation Worksheet

Use this simple worksheet to compare other SSH packages to NanoSSH. Make your own notes right here.



Criteria	Mocana NanoSSH™	Other SSH Package
Code footprint?	70KB	
Impact to footprints when security is added in phases	Yes. A common crypto code base is re-used by multiple Mocana protocol products.	
Easily turn features on and off?	Yes. Features can be turned on and off as needed to build different product versions for multiple price-points.	
Easily build debug and production builds?	Yes. Compilation flags to build debug builds with options to control debug output. Helps creating debug builds for tracking customer-specific problems	
Purpose-built for embedded devices?	Yes. Performance and footprint-optimized for constrained device environments.	
Easily portable across operating systems and processors?	Yes. Available for more than 1,000 OS+CPU combinations.	
Common code base for different operating systems and processors?	Yes. Common code base with cleanly defined abstraction layer for operating system interface makes cross-platform development fast & easy.	
Able to be integrated as a library (zero-threaded)?	Yes. NanoSSH doesn't require an OS, and doesn't place additional task-switching load on CPU when integrated as library.	
Processor specific optimizations available?	Yes. Assembly language optimizations for computationally-intensive operations available off-the-shelf for many processors.	
Guards for preventing recursion attacks?	Yes. Effective utilization of stack by storing messages outside the stack and provides guards to detect and stop recursion attacks	
Off- the-Shelf support for hardware crypto accelerators?	Yes. Supports hardware-based crypto acceleration in the Freescale PowerQUICC "E" CPU product line. Built-in abstraction layer makes for easy integration with other hardware crypto accelerators.	
Asynchronous core?	Yes. Allows the host processor and hardware offload to be fully optimized.	
Support for the latest standards?	Yes. Mocana proactively monitors the latest developments in the IETF & IEEE and rolls out product upgrades frequently.	
Real engineering support available?	Yes. 24/7/365 support—not just for product problems, but also for device development advice and assistance.	
Reliable vendor?	Single vendor offering, comprehensive, integrated suite of security solutions that you can rely on.	

---

## Resource Constraints

Most of today's embedded devices are constrained by limited resource availability. As a result, engineers face quite a challenge when adding functionality to such devices.

### Code Footprint

Mocana security software products are designed to work on resource-constrained devices where a small code footprint is a critical requirement.

NanoSSH can fit into a device with as little as 70KB of available firmware. Other products and open source software take about 5 times the firmware space.

### Heap and Stack Footprints

Heap and stack footprints are important measures of software quality and indications of the attention given to optimizations. Mocana's Nano- products use memory pools to prevent heap fragmentation. They also cap memory utilization and intelligently allocate and release memory.

Some vendors who claim that their solutions have low heap memory utilization are telling only part of the story. Although minimizing heap memory utilization is important, it's even more critical to minimize stack memory utilization. All memory buffer errors can be hostile, but stack overflows can be particularly devastating. If a security solution is stack heavy, you should be concerned that its underlying design is not optimal for embedded devices, which require that stack utilization be kept to a minimum.

Open source products are designed for desktop systems, where virtual and real memory are abundant. Therefore, such products can afford to use generous amounts of memory. But add memory fragmentation and the embedded systems using these products start having difficulties.

---

## Future Resource Requirements

To reduce time to market, device vendors routinely phase security rollouts. Although acceptable to customers, this creates engineering difficulties as footprints grow with every new security piece added. Such piecemeal security implementations can eventually require Megabytes of uncompressed firmware code space.

Mocana Nano- products use a common code base, and are designed so that each additional security piece adds only a very small, incremental overhead when integrated with devices. Resource conservation today will most certainly extend your device's future to add more applications, as well as enabling you to run larger debug builds to solve customer's on-site critical problems.

If you would like to estimate your current and future footprint requirements, please feel free to contact your Mocana Sales Representative or the Mocana Support team. We're confident that Mocana's solutions excel in minimizing future resource requirements.

---

## Code Concerns

This section describes how security solutions address the following common code concerns:

- Feature flexibility
- Portability
- Protection against recursion attacks
- Ease of integration

### Build Flexibility

Diverse product families, varying price points within product families, and the push toward a common code base all add complexity to engineering design and architectural decisions. It can be a challenge to design in the capability to:

- Selectively turn features on and off
- Create both debug and production builds
- Create performance optimized build.

The Mocana Nano- line is designed with this flexibility in mind. Each product can be built just by modifying compilation flags. Likewise, product-specific features can be turned on and off by using compilation flags. By turning unwanted features off you can reduce the foot print. You don't need to rewrite any code, nor keep multiple copies of the code base in your version control system.

NanoSSH provides compile time and configuration options to turn many features on and off, such as port forwarding; synchronous, asynchronous interface, SFTP and various Ciphers. (For a complete list of configurable NanoSSH features, refer to the Mocana DSF Installation Guide.)

### Portability

When we talk about code being portable, we mean code that provides:

- Out-of-the box support for your current and future OS/CPU combinations
- Easy portability to additional OS/CPU combinations
- Code sharing across different OS/CPU-based products

For a security solution to be portable, it must be processor architecture and operating system independent. Open source and many vendor solutions' out-of-box support is limited to PC-based environments. Additional requirements are

---

that the code be flexible with respect to memory alignment, converting between network and host byte order, and that care be taken to prevent bus errors.

Ports based on code that is not CPU-independent often exhibit problems with memory alignment and byte order.

Mocana Nano- products are available for all leading operating systems (Linux, VxWorks, Windows-Mobile, QNX, OSE, and more). In fact, Mocana supports more than 25 operating systems out of the box and more than 1000 OS/CPU combinations. And if you need an additional OS/CPU port, the Mocana abstraction layer makes it easy; ports to new platforms can typically be completed in less than two hours.

### **Protection against Recursion Attacks**

Many security solutions are built without regard to stack memory utilization. Blowing a stack with most embedded operating systems will lead to a crash. Malware writers can exploit such vulnerability by designing worms and denial of service attacks that employ recursion or stack buffer overflows to take over systems and knock them out.

To prevent such attacks, Mocana Nano- code stores messages outside the stack, and includes special guards to detect and stop recursion attacks.

### **Ease of Integration**

Many factors contribute to how easy or difficult is it to integrate security code into your devices and applications:

- Sample code availability (and its quality)
- Documentation

In addition to the expected Release Notes, NanoSSH ships with a full documentation suite: Installation Guide, porting instructions, sample code, and product guide that provide background and detailed best practices, integration instructions, and an explicit HTML-viewable API Reference.

---

## Target System Requirements

When evaluating a potential security solution, it's important to consider your target system requirements:

- Do you need flexibility to integrate security products as a library (zero-threaded)?
- Is the product you're evaluating designed from the ground up and optimized for your target?
- Does the security solution offer optimizations for different processors? Where are the optimizations, and performance benefits?
- Is hardware acceleration needed or will it be required in your future releases?
- Do you require asynchronous operations?

### Threaded and Threadless

Many security solutions are thread-heavy. When every connected client requires a thread or two, context switching time is increased and the solution is not scalable. And worse, such solutions are typically socket blocking, which leads to vulnerabilities such as TCP/IP communication attacks that exhaust the supply of valuable system resources. Security can be CPU intensive and ill-thought out design decisions lead to poor performance.

Mocana Nano- solutions are designed to work equally well in RTOS and non-operating system environments. This key differentiator means that Mocana solutions are scalable and place only a light load on your device's CPU.

### Speed

Some commercial security solutions, while not "open source," are nevertheless derived from OpenSSH. This can lead to performance problems because OpenSSH was simply not designed for embedded systems. Such solutions are missing many features that are critically important to properly protect embedded devices and ensure that they are not subject to race conditions. Knowing this, some vendors are understandably reluctant to disclose the fact that their solutions are OpenSSH-based instead of purpose-built for embedded systems.

Mocana's Nano- products are built from the ground up for embedded systems, and are very fast. For pure software solutions, Mocana has profiled and identified critical CPU-intensive code sections and abstracted them to enable assembly language optimizations. Mocana provides such optimizations for many processors: PPC, MIPS, ARM, 80386+, Coldfire, 68K, and more.

---

## Hardware Acceleration

As silicon vendors increasingly offer crypto offload or hardware acceleration support, it is fast becoming a “must have” for devices to support hardware acceleration—offloading some of the big math operations associated with establishing a secure connection, and performing those operations in hardware.

It is important to understand whether a security solution supports hardware acceleration and how the product is architected to interface with hardware accelerators. Some vendors claim to support hardware acceleration, but it is important to investigate the details to confirm how real their implementations are.

Mocana’s Nano- products use an asynchronous core that allows the host processor and hardware offload to be fully optimized. Sockets can be non-blocking during the full socket lifetime. Data sends and receive functions do not block, which prevents denial of service attacks for communications.

## Asynchronous Operations

An asynchronous core is required for non-blocking sockets, which in turn are required to enable hardware acceleration.

However, many security solutions lack an asynchronous core. The product vendors may still claim that they are capable of non-blocking sockets after an SSL handshake. Unfortunately, most TCP/IP stacks do not allow a socket to switch between non-blocking and blocking.

Mocana’s Nano- products have an asynchronous core that allows the host processor and hardware offload to be optimized fully. Sockets can be non-blocking during the full socket lifetime. Data send and receive functions do not block, which prevents denial of service attacks for communications.

NanoSSH provides compile time options to use either synchronous or asynchronous API functions. NanoSSH also provides API level abstraction to use software crypto libraries or crypto offload to hardware accelerators.

---

## Protocol Standards and Features

Open source projects and low-volume vendor solutions can become “stale,” and fail to maintain support for the full suite of protocol and communications standards (if the code even supported them in the first place). They also frequently fail to fully support all standards and features.

Mocana keeps track of all the latest developments in the IETF, IEEE, and other standard bodies and rolls out product upgrades appropriately. Mocana products are standard-compliant and extensively tested on many different operating systems, processors, and for interoperability issues. Third parties, such as VPNC, provide independent testing and verification for protocol conformance and interoperability.

NanoSSH supports all the latest RFCs. It is fully RFC compliant and interoperable with 3rd party SSH stacks.

NanoSSH provides many features that may not be available in other SSH implementations:

- Group permission protection for read, write per file, directory. In case of directories, parent’s permissions are inherited. This is useful as to prevent a guest/low level user from gaining access to password files, private keys, and other sensitive information.
- SFTP
- SFTP supports per file/directory permission protections even if underlying RTOS does not support it.
- Port forwarding
- Certificate based authentication
- Provides OCSP-based online status check for certificates (Release in spring 2009).

---

## Vendor Services

When choosing a security product vendor, it's important to consider the big picture:

- Who are the vendor's customers?
- Does the vendor offer just a single product or a comprehensive, integrated suite of security solutions?
- What sort of support is available?

Mocana solutions are used by some of the biggest blue chip customers: Dell, Motorola, Nortel Networks, SonusNet, Radvision, AMX, Centillium, Cisco, and more. (See Mocana's website, <http://www.mocana.com/customers.html>, for a list of featured customers.) Many of the biggest device makers have chosen to align or be affiliated with Mocana. Mocana is part of the Freescale Alliance, the Intel Communications Alliance, the MontaVista, WindRiver and QNX Partner Directory.

Mocana is the only vendor that provides a breadth of security products within a single Device Security Framework. All Mocana products come with Mocana's 24/7/365 support—not just for product problems, but also for device development advice and assistance.

NanoSSH is deployed in diverse market segments such as datacomm, aviation entertainment, military, industrial automation, and security solutions.

---

## Conclusion

We hope you've found this document helpful while evaluating NanoSSH. If you have any questions or suggestions for future versions of this document please feel free to drop us a line at [support@mocana.com](mailto:support@mocana.com). We look forward to welcoming you to the Mocana family.