Introduction

DNS is both important and implemented poorly.

IRONSIDES is an Ada/SPARK DNS implementation intended as proof-of-concept:

Can DNS server be implementation on a modest (!!!) development budget with provable security properties?

What does adding provable security properties do to performance?
Milestone 1: Authoritative server on Linux [2012]
BIND and IRONSIDES performance -- Linux

![DNS server performance chart](image)
Milestone 2: Authoritative server under Windows (2013)

DNS server performance (queries/ms)

- BIND
- IRONSIDES
- Win DNS
Test bed
Test bed
max queries per second
(authoritative, Unix)
max queries per second
(authoritative, Windows)
queries sent vs max queries per second
(recursive, Windows)
% queries lost vs max queries per second
(recursive, Windows)

![Graph showing % queries lost vs max queries per second for BIND, IRONSIDES, MARADNS, and WINDNS. The graph indicates that as the number of queries increases, the percentage of queries lost also increases, with MARADNS showing the highest percentage of queries lost.](image-url)
Insights from experience

Performance doesn’t need to be sacrificed to improve security. In fact, there are instances where SPARK and formal methods make faster code.

Conversely, instances where SPARK requirements slow things down, must use manual overrides.

Role of compiler important!

Tools are impressive and humbling.
Provable security properties in IRONSIDES

• No classic buffer overflow
• No incorrect calculation of buffer size
• No improper initialization
• No ineffective statements
• No integer overflow/wraparound
• No information leakage
• All input validated
• No allocation w/o limits (no resource exhaustion)
Provable security properties in IRONSIDES

• No improper array indexing
• No null pointer dereferencing
• No expired pointer dereferencing (use after free)
• No type confusion
• No race conditions
• No incorrect conversions
• No uncontrolled format strings
• All loops guaranteed to terminate
Conclusions

• Comparable performance
• Far fewer resources, lower development costs
• Why similar approaches not more widely adopted?
• Upgrade IRONSIDES to more current version of SPARK
• Develop web server with similar properties
• Formal methods and SCADA, performance