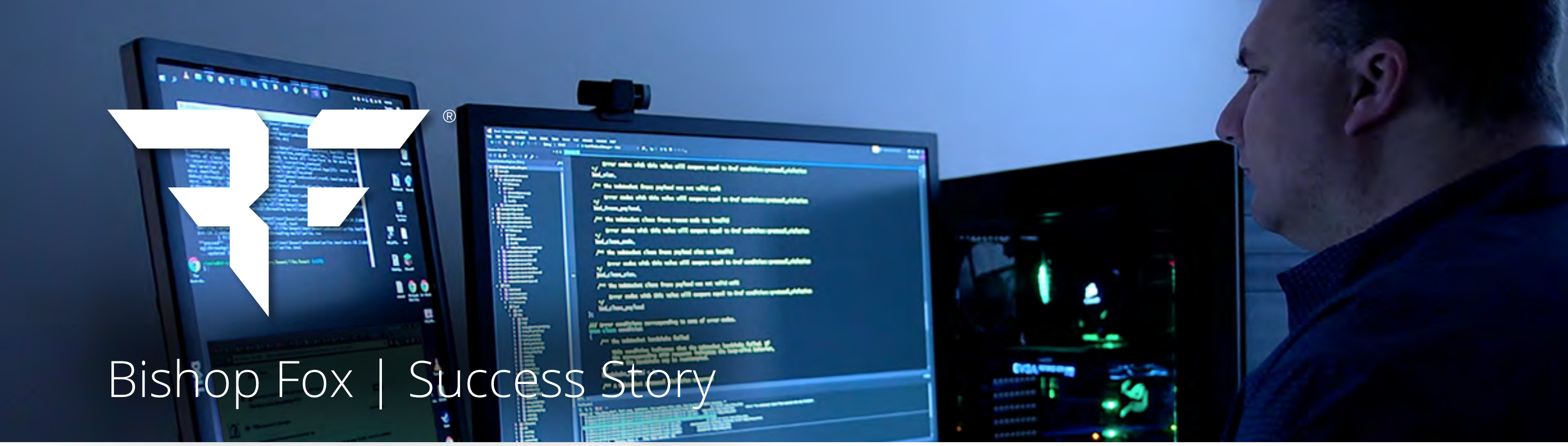

Bishop Fox | Success Story

SECURING BOOST.BEAST - A NON-TRADITIONAL SOURCE CODE REVIEW

Securing the Foundation of Thousands of Web Applications

About Boost.Beast

Beast is an open-source C++ header-only library serving as a foundation for writing interoperable networking libraries. It implements low level HTTP/1, WebSocket, networking protocol vocabulary types, and algorithms using the consistent asynchronous model of Boost.Asio. Beast empowers users to create their own libraries, clients, and servers.

The Challenge

Since tens of thousands of users across the Internet rely on the Beast library as the foundation of their code, security and peer review is crucial and mandatory to identify and remove dangerous security vulnerabilities. Vinnie Falco, the creator of Beast, reached out to Bishop Fox to assess the security of the Boost C++ Beast HTTP/S networking library.

The assessment team conducted a hybrid application assessment of the Beast library. Bishop Fox used its hybrid application assessment methodology to conduct automated scans of the deployed application and source code, detailed inspection of the scan results and manual code review to thoroughly identify potential application security vulnerabilities.

In addition, the Bishop Fox team reviewed the application architecture and business logic to locate any design-level issues. Finally, the team performed manual exploitation and review of these issues to validate the findings.

HIGHLIGHTS

CUSTOMER	Vinnie Falco, Founder and President of C++ Alliance; Creator of Beast
WEBSITE	https://github.com/boostorg/beast
ORGANIZATION TYPE	C++ header-only library
SERVICES PROVIDED	Application and source code security assessment

SUMMARY

- Beast engaged Bishop Fox to assess the security of their C++ code library
- Bishop Fox's hybrid application assessment methodology was used alongside targeted source code review and fuzz testing
- Multiple "high-risk" denial-of-service vulnerabilities were identified and fixed prior to code release
- Beast's engagement with Bishop Fox reflects the open-source project's commitment to security and transparency
- Detailed report of the findings is available here: goo.gl/ZFWW4e

Scan Qr code to access the report

The project was conducted with the following goals:

- Identify critical- and high-risk issues (especially memory corruption issues) in the Beast library that could be exploited to subvert expected application behavior
- Review Beast for security vulnerabilities with a focus on those documented by OWASP and specific to web application technologies
- Determine whether the design of the Beast library meets secure-by-design principles
- Thoroughly fuzz test the example advanced server Beast library application
- Perform a manual review of Beast library application source code to uncover subtle implementation issues that may impact library security

> *“Bishop Fox’s reputation in the industry is exceptional. This project was uniquely challenging as it did not fit the typical profile and scope of an application pen-test. Despite the challenges, Bishop Fox was extremely professional and produced great results.”*

The Results

The Bishop Fox assessment team discovered multiple “high-risk” denial-of-service vulnerabilities that could be exploited by malicious hackers to prevent authorized users from accessing the resource.

The team demonstrated three denial-of-service attacks against Beast by sending malformed WebSocket frames containing a compressed payload. The issues were identified by fuzzing the WebSocket server code responsible for uncompressing client messages.

In addition, the Bishop Fox team found that Beast uses an insufficient source of entropy as a seed value to a linear congruential generator (LCG) to generate random values that serve as the masking value when WebSocket client frames are sent. In special circumstances, an attacker may be able to exploit this issue to poison HTTP caches served from improperly implemented intermediaries.

Bishop Fox produced a detailed report of the findings, outlining where security vulnerabilities could potentially affect developers when using Beast code as foundation. The crash in WebSocket frames vulnerability was fixed in the first official release of Beast in Boost thanks to the Bishop Fox discoveries. The other vulnerability was found to only affect websocket clients in very limited circumstances. Because Beast places immense value on security and transparency, the report and details of the findings were publicly posted on the Beast Project page.

> *“It’s important to me that any developer using Beast knows exactly the strengths and weaknesses of the code they are using. Transparency is extremely important, and the Bishop Fox report demonstrates the high level of security inherent in the Beast library as well as best practices and information for any developer utilizing the Beast code.”*

Vinnie Falco, Author of Boost.Beast

About Vinnie Falco

Vinnie Falco started programming on an Apple II+ in 1982. He wrote BearShare - a Gnutella compatible file sharing program and later joined Ripple, a global financial settlement network built on top of a decentralized cryptocurrency and its associated ledger. Ripple gave him the opportunity to develop Beast, the HTTP and WebSocket library written in C++ and used in Ripple.

About Bishop Fox

Bishop Fox is a global security consulting firm providing penetration testing, web app assessments, and risk mitigation services to help businesses protect data and assets from hacker attacks. We provide actionable cybersecurity guidance to the Fortune 1000, high-tech startups, and financial institutions worldwide.

You can follow us on Twitter @bishopfox.

BISHOP FOX®