Berend-Jan Wever

@ berendj@nwever.nl (PGP)
linkedin.com/in/skylined
twitter.com/berendjanwever
github.com/SkyLined
skylined.nl

Introduction

Berend-Jan Wever is an Offensive Information Security Researcher who specializes in research, pen-testing, fuzzing and automation, vulnerability and risk analysis, and Proof-of-Concept exploit development. His focus has been on finding security issues in threat models, designs and implementations of systems. He has reported of hundreds of security issues in Operating Systems, Web Browsers, Web Servers, and many other products for many different high-profile vendors, impacting billions of end-users. He has helped move the industry forward by publishing various novel attack vectors, exploitation techniques, and mitigation bypasses.

Two decades of experience working with dedicated security team at some of the largest companies in the industry allows Berend-Jan to see the bigger picture and predict the next move. He takes into account the entire system, not just individual components. He will chain together minor issues and design quirks across multiple systems to provide a holistic threat assessment. He determines the maximum impact an advanced threat actor might achieve, where the whole is greater than the sum of its parts.

He looks for patterns in the vulnerabilities found in systems to group them in classes and finds the root cause for each class. He develops automation, suggests process changes, and creates training to help the development team remove every instance of such a class from the product, and prevent re-introduction, to eliminate threats completely. He focusses on the long term, thinking ahead and planning for the next few years. His develops solutions on developing solution that scale, so they can keep up with relentless growth and do not become a bottleneck for productivity.

Berend-Jan creates actionable reports and presents these to the development team to allow them to address issues promptly and completely. He also provides high level overviews for management to assess risk and overall state of security. He advises the development team on how to fix issues and what mitigations to consider to reduce the potential impact of vulnerabilities. He relishes in the opportunity to work less experienced team members to help them get the maximum out of their unique potential and help them grow in their career. He enjoys giving presentations about his work and discussing new ideas to share information and foster innovation.

Technology

Over the decades, Berend-Jan has reviewed projects that use a very long list of different programming languages, protocols, data and file formats, and systems. He has developed programs and scripts using many of these as well. The full list is hard to reconstruct, so only the commonly useful ones are mentioned below.

Operating Systems and Cloud

Microsoft Windows, Linux, Amazon Web Services (AWS).

Languages

Java, Python, PHP, Perl, JScript/JavaScript/ECMAScript/TypeScript, C, C++, C#, IA32 and AMD64 assembly, SQL, Bash, Batch, PowerShell.

Protocols

DHCP, DNS, FTP, HTTP, MQTT, TCP/UDP, IP, ARP, and a long list of custom/internal application layer protocols.

File/Data Formats

Web: XML, HTML, XHTML, SVG, CSS, RSS, JSON, YAML
Media: ANI, BMP, GIF, ICO, JPEG, RIFF, WAV
Documents: DOC, DOCX, PDF, RTF
Windows Binaries: DLL, EXE

Recent Work History

Senior Security Engineer in Holistic Testing Team at AWS

Company: **Amazon Web Services (AWS)**
Location: **Work from home in the Netherlands**
Period: **2021-2024**
Position: **Security Engineer III, L6**
Activities: **Tech lead**, **process management**, **training**, **pen-tests**, **design & implement automation**, **reverse engineering**, **job**

applicant interviews

Within the Holistic Testing Team, Berend-Jan led 4-5 person teams on 2-3 months security reviews and testing of various AWS Services. As one of the most senior engineers on the team, he represented the engineers in discussions with manager about ongoing activities, and short- and long-term planning.

As a tester and testing lead, he performed the follow activities:

1. Collecting of information relevant to the security of the Service, including a review of the Threat Models, identification of critical components and potential attack vectors, and ranking identified risks by potential impact and feasibility.
2. Discuss under-documented components with the Service Team and/or reverse-engineer systems to create the required documents and store everything in a single, well-structured location for future use.
3. Developing a test plan that fits the team's size, expertise, schedule and takes into account the level of support the Service Team is capable of providing, as well as distributing the required tasks among the team members.
4. Using the collected information to focus on the most vital areas for the security of the Service and create a targeted pen-testing scope that covers the most relevant areas.
5. Working with the Holistic Testing team members and the Service Team to perform pen-tests of scope-items to identify, analyze and report security vulnerabilities. Helping the Service Team design and deploy fixes and mitigations for the identified risks. Adjusting the scope as needed when a new high risk area outside our scope was identified during testing.
6. Designing and implementing tools to help automate parts of the pen-testing process, such as scanning and collecting information about the Service's components, automated testing of components, fuzzing, and the creation of Proof-of-Concept exploits.
7. Identifying trends/patterns in security issues to detect their underlying root-causes and work with the Service Team. Based on this, the team would be advised to go through training to address knowledge gaps, use tools to detect or prevent issues, consider system design changes, and/or process changes to reduce/remove the risk of introducing more similar issues in the future.
8. Writing and presenting a report both detailing the technical results of our tests for engineers as well as providing a high-level overview of our findings for management. These reports always included suggestions for future areas of improvement and future security reviews.

Berend-Jan worked to standardize the design and implementation of the team's in-house tools, as well as their output formats with the aim of making the tools more compatible/interoperable. He introduced a generic modular design to make the tools easier to maintain and expand with new features. This included tools for scanning text for sensitive or security-relevant information, tools to scan for and detect configuration issues that have security impact, and tools to test specific components and/or confirm the existence of theoretical issues (aka. Proof-of-Concept code).

As a security-minded user, he identified security issues in internal systems and processes that he came in contact with, and worked with the relevant team to explain the potential impact, assess priority, and help create a plan to address the issue in a reasonable time-frame. He worked with the team to make sure that the implemented changes completely removed the threat.

Berend-Jan reviewed and improved internal security training for developers. He presented on various security topics to explain how he approaches certain problems, how he designs tools to be generic, modular, and interoperable, and how he discovered various security issues and worked with development teams to help them address issues quickly, and how to avoid causing friction to make sure security engineers and developers cooperate effectively.

Offensive Security Researcher and head of Fuzzing Community at Intel

Company: **Intel**
Location: **Work from home in the Netherlands**
Period: **2019-2021**
Position: **Senior Security Researcher**
Activities: **Tech lead**, **training**, **design & implement automation**

Berend-Jan set up and led a cross Intel effort to automate security research with a strong focus on fuzzing. Through meetings and presentations, he strived to adapt specialized custom tools to test hardware, firmware and drivers to make them more generic and useful to others teams, raising awareness of their existence and improving their documentation to foster adoption across Intel. He connected people with similar goals and ideas to work together and achieve results faster.

He architected a modular framework for running automated tests on arbitrary targets, both virtual and running real hardware, and detect and analyze issues triggered through these tests, group and de-duplicate similar issues, and create actionable reports. He worked with various teams across Intel to refine this design to make sure it would be easy to get started, possible to fuzz notoriously hard-to-fuzz products with minimal efforts, and requiring limited or no knowledge of fuzzing to do so. The design continuous fuzzing to start early in development and continuing after the product has been released.

- Make it easy to implement custom fuzzers for specific features, and share these with teams developing other products with similar features.
- Allow the collection of meaningful information about fuzzer deployment that can be used to assess effectiveness and check SDL requirements.
- Allow security researchers to easily create, test and deploy new fuzzing engines.

Owner of SkyLined Security

Company: **SkyLined Security**
Location: **Work from home in the Netherlands**
Period: **2011-2019**
Position: **Owner**
Activities: **fuzzing**, **design & implement automation**, **Proof-of-Concept development**, **pen-tests**, **reverse engineering**

- Developed automated security tests (fuzzers) that found a large number of issues in high-profile targets, including Microsoft Windows, Microsoft Internet Explorer, Microsoft Edge, Google Chrome, and Mozilla Firefox.
- Reported a large enough number of security issues to Microsoft to be included in the Microsoft Security Research Center Top 100 Security Researchers list 3 years in a row starting from its inception in 2015.
- Worked as an external consultant on a wide-range of projects from pen-testing, code-, and design-reviews, through fuzzer development and integration to writing whitepapers and guidance documents.

Security Researcher in Chrome Security Team at Google

Company: **Google**
Location: **Work from home in the Netherlands**
Period: **2008-2011**
Position: **Senior Software Security Engineer**
Activities: **pen-tests**, **design & implement automation**, **fuzzing**, **developing patches and mitigations**

Hired as an initial member of the Google Chrome Security Team. Hit the ground running to make sure the software did not ship with any major security issues three months later.

- Developed tools to look for and analyze security issues in nearly all parts of the codebase, which are still actively used to this date.
- Found large numbers of vulnerabilities, analyzed externally reported vulnerabilities and guided the patch process to release comprehensive fixes to customers at record speeds.
- Improved security by mitigating or preventing exploitation of issues through design and implementation changes and offered guidance and advise to Chromium project members on security related topics.
- Participated in setting up and maintaining the Google Chrome Vulnerability Reward Program.
- Developed fuzzers that are still being used and finding issues in Google Chrome today.

Security Researcher in Security Windows Initiative Attack Team at Microsoft

Company: **Microsoft**
Location: **Work from home in the United Kingdom**
Period: **2005-2008**
Position: **Security Software Engineer**
Activities: **pen-tests**, **design & implement automation**, **fuzzing**, **analyzing external security reports**, **checking patches**, **job interviews**

- Joined a newly formed team of two security researchers working from home in Cheltenham, UK, as the European branch of the larger SWI-AT team.
- Developed better tools to automate the detection of security issues, such as fuzzers and compiler plug-ins that detect vulnerable design and implementation patterns.
- Reviewed code and patches and analyzed external reports of vulnerabilities in Microsoft products.
- Researched new attack vectors and techniques and shared knowledge with developers working on various product teams.
- Took part in the hiring and on-boarding of additional team members in the UK.

Skills and Experience

Key skills

- Experience in security on a wide range of products, including hardware, firmware, drivers, server software, client software and websites, as well as process improvement and security education.
- Individual contributor and tech lead. Enjoys guiding junior team members to help them exploit their strengths and avoid/improve their weaknesses. Loves explaining techniques and tricks and giving tips on how to find, analyze and address security weaknesses to anyone who will listen.
- 20 years of experience in fuzzing, both in creating fuzzers, using them to find issues and processing the results to improve the security of a wide range of products. Knows how to avoid overwhelming a development team with new findings by prioritizing and filtering based on severity/impact and development team size.
- Thinks far outside the box and has a history of published innovative and creative security techniques to prove it.
- A wide range of projects has provided experience in a large number of topics. From user-land applications to operating system kernels, firmware, and hardware. From design to implementation and configuration reviews, in a wide-range in programming, scripting and markup languages. From client to cloud, through front-end, back-end, proxies and servers, as well as the network layer. From protocols and file-formats to memory layout and CPU features.
- Comfortable and experienced with learning new programming languages, applications, frameworks and systems on the job. The only think I have avoided is cryptography.

Historically Notable Publications

A small selection of contributions to the information security community:

- BugId is a Python script that automatically analyzes crashes and determine their security impact. It reduces the need for a large and costly team of specialized security engineers. It reduce time-to-patch and window of exploitability by prioritize important issues and speeding up analysis. It can create detailed reports containing both concise management-level information and extensive technical details.
- Browser Security Whitepaper that collects all relevant information on the security of a number of different web browsers. It covers everything from high-level configuration management to low-level security technologies. IT managers can use it to make an informed decision about which browser is best suited for their specific needs. It offers security experts evidence based data on which browser protects best against specific risks.
- Released technical analysis of a large number of vulnerabilities and techniques.
- Illustrated the fragility of state-of-the-art web browser code in 2016 through daily Tweets about a new way to crash a web browser.
- Introduced heap-spraying in web browsers, a technique that facilitates exploitation of vulnerabilities in application. This technique has been widely adopted and built upon to bypass mitigations and create sophisticated and reliable exploits.
- Main author of the world's smallest Windows shellcode.
- Inventor of the concept of Omelette shellcode.
- Creator of the first practical alphanumeric shellcode encoder, which was ported to the Metasploit framework and author of ASCII art shellcode.
- Created the first Proof-of-Concept XSS worm in 2002 to warn of their potential danger; the first publicly released XSS worms proved they can causing serious damage 3 years later.

Spoken languages

Dutch - native
English - Fluent
German - Proficient
French - Basic