Berend-Jan Wever

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

Senior Information Security Researcher

- 20 years of professional experience in security, including working with specialist security teams at Microsoft, Google, Intel and Amazon Web Services (AWS).
- Extensive experience with security research, pen-testing, fuzzing, developing automation (including targeted fuzzers), Proof-of-Concept exploit development, and as a technical lead.
- Designer of various novel attack vectors, exploitation techniques, and mitigation bypasses, such as XSS-worms, heap-spraying, and alphanumeric shellcode encoders.
- Discovered and reported hundreds of security issues in Operating Systems, Web Browsers, Web Servers, and many other products for many different vendors, impacting billions of end-users.

Berend-Jan has made his career in offensive security. His work focuses on looking for holes in the threat model, design and/or implementation of systems. He creates actionable reports for the development team to address these issues promptly, and provides high level overviews for management to assess overall state of security. He advises the development team on how to fix such issues and what mitigations to consider to reduce the impact of such vulnerabilities. He constantly looks for patterns in the vulnerabilities in a product to group them in classes. He will find the root cause and symptoms for each class. Through automation, process changes, and training he helps the development team remove every instance of such a class from the product, to eliminate the threat completely.

Extensive experience allows him to sees the bigger picture and determine the maximum impact an advanced threat actor might achieve. He can chain together issues and design quirks across a system to provide a holistic threat assessment that takes into account the entire system, not just individual components. He focuses mainly on the long term, thinking ahead and planning for the next few years.

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 interviews**

Within the Holistic Testing Team, I led 2-3 months security reviews and testing of AWS Services aimed at helping the Service Team improve their security in the long term. I took ownership of (re-)designing and developing various tools to help automate our work. I reported security issues in other system and processes I came into contact with, outside of my day-to-day work to improve overall security of AWS. As a senior engineers on the team, I represented the engineers with management when discussing our teams ongoing activities, and short- and long-term planning. I gave various presentations on security related topics to educate others within AWS and helped review and rewrite security training material.

As a testing lead, I performed the follow activities:

1. I distributed the tasks of collecting information relevant to the security of the service, reviewing the Threat Model for the service, identifying critical components and potential attack vectors, and ranking identified risks by potential impact and feasibility. Where this information was not yet fully available or well organized, I made sure it was collected in a single location for future use.
2. I used the collected information to select the most important areas for security and translate these into a targeted pen-testing scope, with each scope item assigned to one or more individual team members for a specific duration, taking into account the skills and interests of individual team members.
3. I worked with the my team and the service team to perform pen-tests of scope-items to identify, analyze and report security vulnerabilities. We helped the service team design and deploy fixes and mitigations for the identified risks. I adjusted the scope as needed when we identified a new high risk area outside our scope during testing.
4. I designed and implemented 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 to create Proof-of-Concept exploits.
5. Identify trends/patterns in security issues to detect deeper root-causes and work with the team to select training to address knowledge gaps. Help the team select or design tools, consider system design changes, and/or process changes to reduce/remove the risk of introducing new issues.

6. Write a report detailing the technical results of our tests as well as a high-level overview of our findings, including suggestions for future areas of improvement and future security reviews.
7. Present the results to Senior management of the service.

For our tools, I worked to standardize their design and implementation, and their output format(s) with the aim of making the tools compatible/interoperable. I 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, I identified security issues in internal systems and processes that I used, 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. I made sure that the implemented changes completely removed the threat.

I reviewed and improved internal security training for our developers. I gave various presentations on security related topics to explain how I approached certain problems, how I designed various tools, how I discovered various security issues, how to work with development teams to help them address issues quickly, and 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**

Leading the effort to automate security research with a strong focus on fuzzing. Many of Intel's products are hardware based with custom firmware and drivers, which do not easily lend themselves to fuzzing with existing tools. The existing tools are often under-documented, hard to setup and deploy at scale for anyone who is not a security researchers with experience in fuzzing. Choosing which fuzzers to apply, how to apply them correctly and collect meaningful and actionable information about their effectiveness is currently too complex. I am leading an effort to create a modular fuzzing framework that resolves these issues. This project has multiple goals:

1. Make it easier for product teams to start effectively using continuous fuzzing during development to catch security issue early, without requiring these developers to have a thorough understanding of fuzzing and the various possible techniques/engines that can be deployed.
2. Make it easier to fuzz anything, including currently hard-to-fuzz products such as hardware and firmware. From the start of development all the way through the development process and continuing after the product has been released.
3. Allow the collection of meaningful information about fuzzer deployment that can be used to implement and check SDL requirements.
4. 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