Berend-Jan Wever

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

Introductie

Berend-Jan Wever is een Offensieve Information Security Researcher gespecialiseerd in onderzoek, pen-testing, fuzzing en automatisering, beveiligingsfout en risico analyse, en "Proof-of-Concept exploit" ontwikkeling. Zijn focus ligt op het vinden van beveiligingsfouten in "Threat Models", ontwerp en implementatie van systemen. Hij heeft honderden beveiligingsfouten gerapporteerd in Besturingssystemen, Web Browsers, Web Servers, en vele andere producten van vele verschillende grote aanbieders, die impact hadden op miljarden eindgebruikers. Hij heeft de beveiligingsindustrie vooruit geholpen met publicaties over nieuwe aanvalstechnieken, exploit-technieken, en manieren om mitigations te omzeilen.

Twee decennia werkervaring als beveiligingsexpert voor enkele van de grootste bedrijven in de industrie, geven Berend-Jan de mogelijkheid om het grote plaatje te zien en de volgende stap te voorspellen. Hij kijk naar het hele systeem, niet alleen de individuele onderdelen. He rijgt kleine problemen en ontwerp-eigenaardigheden aan elkaar uit meerdere systemen om een totaalanalyse te geven van de risico's. Hij bepaald de maximale impact die een geavanceerde aanvaller zou kunnen hebben, waarbij het totaal groter is dan de optelsom van de onderdelen.

Hij zoekt naar patronen in gevonden beveiligingsfouten, groepeert deze in klassen, en zoekt de bron van het probleem voor elke klasse. Hij ontwikkeld automatisering, suggereert process veranderingen, en creÃ«ert training om het ontwikkelteam te helpen alle exemplaren van zo'n klasse te verwijderen, en om te voorkomen dat deze opnieuw worden geÃ¯ntroduceerd. Hij focust op de lange termijn, denkt vooruit and maakt plannen voor de komende paar jaar. Hij ontwikkeld oplossingen die mee kunnen schalen met enorme groei en geen belemmering vormen voor productiviteit.

Berend-Jan creÃ«ert rapporten waaruit direct actie kan worden ondernomen door het ontwikkelteam, zodat zij de problemen direct en compleet op kunnen lossen. Hij voorziet management van een samenvatting waarmee risico's en de algemene staat van beveiliging kan worden bepaald. Hij helpt het ontwikkelteam bij het oplossing van beveiligingsproblemen en adviseert over mitigations die overwogen kunnen worden om de potentiÃ«le impact van beveiligingsfouten te verminderen. Hij geniet van de mogelijkheid om met minder ervaren team leden te werken, en hen zo te helpen het maximale uit zichzelf te halen en te groeien in hun carriÃ¨re. Hij geeft graag presentaties over zijn werk en houdt van discussies over nieuwe ideeÃ«n om zo informatie te delen en innovatie aan te wakkeren.

Technologie

Over de decennia heeft Berend-Jan gewerkt met projecten waarbij een heel lange lijst aan verschillende programmeertalen, protocollen, data- en bestandsformaten, en systemen. Hij heeft in velen hiervan ook programma's en scripts ontwikkelt. De volledige lijst is lastig op te stellen, daarom worden hieronder alleen de vaak gebruikte genoemd.

Besturingssystemen en Cloud

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

Talen

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

Protocollen

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

Data- en Bestandsformaten

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

Recente Werk Geschiedenis

Senior Security Engineer in Holistic Testing Team bij AWS

Bedrijf: **Amazon Web Services (AWS)**
Locatie: **Werk vanuit huis in Nederland**
Periode: **2021-2024**
Positie: **Security Engineer III, L6**
Activiteiten: **Technisch leidinggevende**, **process management**, **training**, **pen-tests**, **ontwerp en implementatie van automatisering**, **reverse engineering**, **sollicitatie interviews**

In het "Holistic Testing Team" leidde Berend-Jan teams van 4-5 personen tijdens 2-3 maanden durende beveiligingsreviews en tests van diverse AWS Services. Als een van de meest seniore engineers in het team vertegenwoordigde hij de engineers richting het manager in discussies over de activiteiten en de korte- en lange-termijn planning.

Als tester em test teamleider heeft hij de volgende activiteiten verricht:

1. Verzamelen van informatie die relevant is voor de beveiliging van de Service, zoals een beoordeling van de Threat Models, identificatie van de meest kritieke onderdelen en de potentiÃ«le aanvalspaden, en het rangschikken van de geÃ¯dentificeerde risico's op grond van potentiÃ«le impact en haalbaarheid.
2. Bespreken van onder-gedocumenteerde onderdelen met het Service Team en/of reverse-engineeren van systemen om de benodigde documentatie te verkrijgen en alles opslaan op een specifieke locatie op een gestructureerde manier voor gebruik in de toekomst.
3. Ontwikkelen van een test plan dat past bij de grootte van het team, de expertise ven de team leden, de beschikbare tijd. Hierbij wordt rekening gehouden met het verwachte niveau van ondersteuning door het Service Team, en de verdeling van de vereiste taken onder de verschillende teamleden.
4. Gebruik maken van de verzamelde informatie over de service om de focus te leggen op de meest cruciale gebieden voor de beveiliging van de Service en zo een gericht pen-testing plan te maken dat alle relevante gebieden dekt.
5. Samenwerken met de leden van het Holistic Testing team en het Service Team om de pen-tests van alle onderdelen in het plan uit te voeren, om zo beveiligingsfouten te identificeren, analyseren en rapporteren. Het Service Team kreeg hierbij onze hulp om oplossingen te ontwerpen en uitvoeren en mitigations in te voeren voor de geÃ¯dentificeerde risico's. We pasten eventueel ons plan aan als dat nuttig was omdat we nieuwe ernstige risico's identificeerden die buiten het originele plan vielen.
6. Ontwikkelen en implementeren van Tools om delen van het pen-test process te automatiseren, zoals scannen en verzamelen van informatie over de componenten van een Service, het automatiseren van testen van componenten, fuzzing, en het maken van "Proof-of-Concept" exploits.
7. Identificatie van trends en patronen in beveiligingsproblemen om zo onderliggende oorzaken te achterhalen. Gebaseerd op deze informatie werd het Service Team geadviseerd training te doorlopen om gaten in kennis te vullen, tools te gebruiken om fouten te detecteren of voorkomen, hun systeemontwerp aan te passen, en/of hun processen aan te passen om hierdoor het risico dat soortgelijke fouten in de toekomst weer geÃ¯ntroduceerd worden te verkleinen of verwijderen.
8. Schrijven en presenteren van een rapportage met zowel technische resultaten van onze tests voor engineers als een hoog-niveau overzicht van onze bevindingen voor het management. Deze rapporten bevatten altijd suggesties oim de beveiliging nog beter te maken en voor toekomstige tests.

Berend-Jan heeft gewerkt aan het standaardiseren van het ontwerp en de implementatie van de tools die het team ontwikkeld voor eigen gebruik. Ook heeft hij het formaat waarin ze hun resultaten opslaan gestandaardiseerd om zo de tools beter samen te laten werken. Hij heeft een generiek modulair ontwerp geÃ¯ntroduceerd om deze tools makkelijker onderhoudbaar en uitbreidbaar te maken. Het ging hier onder andere om tools om text te scannen voor gevoelige of relevante informatie, configuraties te controleren op instellingen die een impact kunnen hebben op de beveiliging, en tools die bedoelt zijn om specifieke componenten te testen en/of het bestaan van specifieke problemen aan te tonen (zgn. "Proof-of-Concept" code).

Als een gebruiker van interne systemen en processen identificeerde Berend-Jan meerdere beveiligingsfouten en werkte samen met de relevante teams om het probleem duidelijk te maken, een oplossing te bedenken, de prioriteit van deze oplossing in te schatten, en zo een plan te maken om de problemen te verhelpen binnen een acceptabele tijdsduur. Hij hielp het team te controleren of de doorgevoerde verbeteringen ook daadwerkelijk alle problemen verhielpen.

Berend-Jan heeft interne trainingen voor software ontwikkelaars gecorrigeerd en uitgebreid. Hij heeft presentaties gehouden over diverse beveiligingsonderwerpen waarin hij uitlegt hoe hij bepaalde problemen aanvaart, en hoe hij zijn tools zo generiek, modulair, en interoperabel mogelijk maakt. Ook legde hij uit hoe hij diverse beveiligingsproblemen heeft gevonden en ervoor gezorgd heeft dat deze snel en op de juiste manieren opgelost werden, door frictie met het ontwikkelteam te voorkomen om zo beveiligingsmedewerkers en ontwikkelaars efficient te laten samenwerken.

Offensive Security Researcher en hoofd van de Fuzzing Community bij Intel

Bedrijf: **Intel**
Locatie: **Werk vanuit huis in Nederland**
Periode: **2019-2021**
Positie: **Senior Security Researcher**
Activiteiten: **Technisch leidinggevende**, **training**, **ontwerp en implementatie van automatisering**

Berend-Jan startte and leidde een focus group binnen Intel om beveiligingswerk te automatiseren met een sterke focus op fuzzing. Hierbij werd gekeken naar de mogelijkheden om gespecialiseerd interne tools om hardware, firmware en drivers te testen meer generiek en herbruikbaar te maken, met als doel dat meer teams makkelijker hun producten konden testen. Onderdeel hiervan was het bekend maken onder de teams van de beschikbare tools, het verbeteren van de documentatie en gebruikersvriendelijkheid, en de onderlinge compatibiliteit van de verschillende tools. Het einddoel was een overkoepelend modulair framework om geautomatiseerde tests te kunnen uitvoeren op alle soorten producten van Intel, of deze nu op gevirtualiseerde, gesimuleerde, of werkelijk hardware draaiden. De doelen van dit framework waren:

1. Het makkelijk maken om te starten met fuzzen/testen, onafhankelijk van welk product getest moet worden. Dit gold met name voor producten die op dat moment lastig te testen waren, zoals nieuwe hardware en firmware. Dit moest mogelijk worden zonder dat de gebruiker veel ervaring hiermee nodig had.
2. Het mogelijk maken om producten continue te testen tijdens het ontwikkelen, en deze tests door te laten gaan ook nadat

het product al op de markt was.
3. Het makkelijk maken om een test of fuzzer te schrijven voor specifieke features, data formaten, of protocollen van een product, en deze tests beschikbaar te maken voor alle teams binnen Intel die vergelijkbare tests kunnen gebruiken.
4. Het verzamelen van nuttige informatie over de uitgevoerde tests, om zodoende de effectiviteit te kunnen meten en te controleren of er aan de SDL (Software Development Life-cycle) eisen is voldaan.
5. Het voor security medewerkers mogelijk maken om gemakkelijk nieuwe fuzzing-engines toe te voegen en deze op alle beschikbare doelen te laten draaien.

Eigenaar van SkyLined Security

Bedrijf: **SkyLined Security**
Locatie: **Werk vanuit huis in Nederland**
Periode: **2011-2019**
Positie: **Eigenaar/ZZP-er**
Activiteiten: **fuzzen**, **ontwerp en implementatie van automatisering**, **Proof-of-Concept exploit ontwikkeling**, **pen-tests**, **reverse engineering**

- Berend-Jan heeft geautomatiseerde beveiligingstests (fuzzers) ontwikkeld welke een groot aantal fouten heeft gevonden in zeer wijd gebruikte software, zoals Microsoft Windows, Microsoft Internet Explorer, Microsoft Edge, Google Chrome, and Mozilla Firefox.
- Er werden hierbij dusdanig veel fouten gemeld aan Microsoft dat hij 3 jaar op rij vermeld werd in Microsoft Security Research Center Top 100 Security Researchers .
- Berend-Jan heeft daarnaast als externe consultant aan een divers aantal projecten gewerkt van pen-testing, code-, en ontwerp-reviews, en het ontwikkelen en integreren van fuzzers, tot aan het schrijven van whitepapers en documentatie.

Security Researcher voor het Chrome Security Team van Google

Bedrijf: **Google**
Locatie: **Werk vanuit huis in Nederland**
Periode: **2008-2011**
Positie: **Senior Software Security Engineer**
Activiteiten: **pen-tests**, **ontwerp en implementatie van automatisering**, **fuzzing**, **ontwikkelen van patches en mitigations**

Berend-Jan werd aangenomen als de eerste persoon binnen het Google Chrome Security Team. Hij moest direct hard aan de slag omdat het product drie maanden later de markt op ging.

- Hij heeft software ontwikkeld om fouten op te sporen en te analyseren in vrijwel elk onderdeel van de code. Een aantal hiervan wordt op de dag van vandaag nog steeds gebruikt.
- Hij heeft een groot aantal fouten gevonden en geanalyseerd, en daarnaast ook fouten die door externe partijen werden gemeld geanalyseerd en afgehandeld. Hierbij werd zorg gedragen dat de patch daadwerkelijk alle varianten van het probleem oploste en binnen recordtijd bij de klant kon worden toegepast.
- Door te werken aan mitigations en verandering in het ontwerp heeft hij het minder makkelijk gemaakt om fouten daadwerkelijk uit te buiten en door advies en uitleg aan andere leden van het Chromium team members, heeft hij de bekendheid en bekwaamheid met beveiliging binnen het team verhoogd.
- Samen met zijn team heeft hij het Google Chrome Vulnerability Reward Program opgezet en beheerd.

Security Researcher voor het Security Windows Initiative Attack Team van Microsoft

Bedrijf: **Microsoft**
Locatie: **Werk vanuit huis in het Verenigd Koninkrijk**
Periode: **2005-2008**
Positie: **Security Software Engineer**
Activiteiten: **pen-tests**, **ontwerp en implementatie van automatisering**, **fuzzing**, **analyseren van beveiligingsfouten**, **controleren van patches**, **sollicitatie interviews**

- Berend-Jan vormde samen met twee Engelse collega's een nieuw team in Cheltenham, zijnde ede Europese tak van het SWI-AT team.
- Hij ontwikkelde software om beveiligingsfouten te detecteren, zoals fuzzers en compiler plug-ins die problematische ontwerp en implementatie patronen rapporteerden voor verdere analyse.
- Hij controleerde code en patches op correctheid en analyseerde externe meldingen van softwarefouten in Microsoft producten.
- Hij onderzocht en ontwikkelde nieuwe aanvalspatronen en technieken en deelde deze kennis met software ontwikkelaars bij diverse product teams.

Kennis en ervaring

- Ervaring met beveiliging van een uitgebreide lijst producten, zoals hardware, firmware, drivers, server software, client software en websites. Daarnaast heeft hij gewerkt aan process verbetering en security educatie.
- Individual contributor en Technisch leidinggevende. Berend-Jan houdt er van om junior teamleden te helpen om hun krachten te gebruiken en hun zwakheden te vermijden/verbeteren. Hij houdt er van om tips en trucks uit te delen over hoe je beveiligingsproblemen vindt, analyseert, en oplost aan wie maar wil luisteren.
- 20 jaar ervaring in fuzzing, zowel in het maken van fuzzers, ze gebruiken om problemen te vinden, het verwerken van de

resultaten, en het melden van de problemen aan de ontwikkelaar. Hij weet dat je een team niet moet overweldigen met een grote stroom fouten, maar dat je deze moet filteren/prioriteren op basis van ernst en impact en de grootte van het ontwikkelteam.
- Denkt ver buiten de begane paden en heeft een lijst een innovatieve en creatieve security technieken gepubliceerd die dat bewijzen.
- een zeer gevarieerde lijst van projecten heeft geresulteerd in een ruime ervaring in diverse onderwerpen. Van user-land applicaties to besturingssysteem kernels, firmware, en hardware. Van ontwerp tot implementatie en configuratie reviews, in uitgebreide lijst programmeer-, script- en markup-talen. Van client tot cloud, via front-end, back-end, proxies en servers, en ook op de netwerklaag. Van protocollen en bestandsformaten tot geheugen-layout en CPU-specifieke functies.
- Berend-Jan heeft ruime ervaring, en geen probleem, met het leren van nieuwe talen, applicaties, frameworks, en systemen tijdens het werk. Het enige onderwerp wat hij vermeden heeft is cryptografie.

Historisch Belangrijke Publicaties

Een kleine selectie van contributies aan de computerbeveiligings wereld:

- BugId is een Python script dat automatisch crashes analyseert en uitzoekt of ze een beveiligingsrisico vormen. Het verminderd de noodzaak om een groot en kostbaar beveiligingsteam hiervoor te hebben. Het vermindert de tijd die nodig is om een fout te analyseren, en maakt het makkelijk om fouten met de hoogst mogelijke impact een passende hogere prioriteit te geven. Dit resulteert in een korte tijd tussen vinden en fixen en daarmee ook de tijd waarin er misbruik kan worden gemaakt van deze fouten. Het produceert gedetailleerde rapporten met hierin een korte management-level uitleg en uitgebreide technische details.
- Browser Security Whitepaper met alle relevante informatie omtrent de beveiliging van een aantal web browsers. Dit omvat alles van configuratie instellingen op hoog niveau tot diepe interne security technologieÃ«n. IT-managers kunnen dit document gebruiken om een goed ingelichte beslissing te nemen over welke browser het beste past bij hun specifieke wensen en eisen. Het biedt security experts op herhaalbare tests gebaseerde data over welke browser het beste bescherming biedt tegen een aantal specifieke risico's.
- Een lijst van publicaties met technische analyses van een groot aantal beveiligingsproblemen en technieken.
- Onderstreepte de fragiliteit van de op dat moment modernste web browsers in 2016 door dagelijkse Tweets met details over een fout die een browser liet crashen.
- Introduceerde heap-spraying in web browsers, een techniek die het uitbuiten van beveiligingsfouten in applicaties vergemakkelijkte. Deze techniek werd wijd verspreidt en gebruikt. Hij ontwikkelde diverse varianten om mitigations te omzeilen en geraffineerde en betrouwbare exploits te ontwikkelen.
- Hoofd auteur van 's werelds kleinste Windows shellcode.
- Uitvinder van het idee achter Omelette shellcode.
- Ontwikkelaar van de eerste praktische alphanumeric shellcode encoder, welke is geÃ¯ntegreerd in het Metasploit framework en auteur van ASCII art shellcode.
- Ontwikkelaar van de eerste Proof-of-Concept XSS worm in 2002, en waarschuwde voor het gevaar. De eerste publieke XSS worms toonde aan dat dit soort wormen inderdaad serieuze schade konden veroorzaken 3 jaar later.

Talen

Nederlands - Moedertaal
Engels - Vloeiend
Duits - Redelijk
Frans - Basis kennis