Secure JavaScript code

JavaScript-code kan kwaadwillig, per ongeluk of onverantwoord worden gebruikt om ongewenste resultaten te produceren. Er zijn echter ook methoden die u kunt volgen om ervoor te zorgen dat de code die u op uw webpagina's maakt en gebruikt, veilig is voor uw gebruikers. Houd bij het ontwikkelen van uw webpagina's rekening met de volgende tips om uw JavaScript zo veilig mogelijk te maken:

Test, test, test uw scripts.

Test ze zelf in verschillende browsers. Zorg ervoor dat elk script werkt zoals verwacht – u zou niets onverwachts moeten zien gebeuren.

Blijf up-to-date in uw kennis over JavaScript en de beveiliging ervan

Gebruik de bronnen die er zijn, lees en blijf op de hoogte. De Slashdot-site (www.slashdot.org) plaatst bijvoorbeeld doorgaans zero-day-exploits, waardoor u de kans krijgt om ze te repareren voordat uw site(s) worden gehackt.

Gebruik geen verouderde code

De W3C beëindigt bepaalde HTML-tags wanneer deze de standaarden bijwerkt, vaak vanwege beveiligingsproblemen. Veel ontwikkelaars blijven verouderde tags gebruiken als tijdelijke oplossing voor oudere browsers, maar deze praktijk belemmert de compatibiliteit en kan uw pagina's kwetsbaar maken voor exploits. Als u verantwoordelijk bent voor een website, moet u de pagina's bijwerken naar HTML5. Netjes geschreven, is de huidige code veruit uw beste verdediging. En wanneer uw HTML up-to-date is, presteert uw JavaScript beter.

Gebruik de juiste coderings- en validatiepraktijken

Zoals u hebt geleerd met cross-site scripting, kunt u uw site – en uw gebruikers – beschermen tegen bepaalde soorten aanvallen door code te maken die aan strikte syntaxisstandaarden en controlepraktijken voldoet. Gebruik de tools van gerenommeerde organisaties om u bij deze taken te helpen.

Zorg dat u de code kent die u gebruikt voordat u deze op een website plaatst

Kopieer en plak niet simpelweg code die belooft te doen wat u nodig hebt – bekijk deze eerst zorgvuldig. U kunt zeer nuttige code op internet vinden, maar u moet begrijpen hoe deze werkt voordat u deze gebruikt.

Schrijf uw code consistent

Schrijf uw code consistent zodat de volgende ontwikkelaar die ermee werkt uw code kan begrijpen en snapt waarom u deze op die manier hebt geschreven. Gebruik bij voorkeur een frontend framework dat actief onderhouden wordt zoals Angular of VueJS of libraries zoals React.

Voorzie uw code uitgebreid van commentaar

Opmerkingen die uitleggen waar of hoe dingen zijn gecodeerd, helpen de volgende ontwikkelaar die met uw code werkt. Als er zich een beveiligingsprobleem voordoet, kunt u het gemakkelijker en sneller vinden in een goed geschreven, goed gedocumenteerde code.

Houd beveiligingspatches up-to-date

Dit geldt niet alleen voor browsers en besturingssystemen, maar ook voor databases. Werk de beveiligingspatches bij als u een database onderhoudt. Veel databasehacks (SQL-injectie genoemd) beginnen met een beveiligingslek in uw paginacode – HTML of JavaScript. Hackers kunnen schadelijke code rechtstreeks in de database plaatsen. Dit maakt de aanval veel moeilijker te ontdekken en op te schonen omdat uw broncode er nog steeds perfect uitziet.

Houd ook uw besturingssysteem up-to-date

Veel computers worden nu verkocht met een vooraf geïnstalleerde, specifieke versie van het besturingssysteem en de koper moest voorheen eventuele besturingssysteemupgrades kopen. Tegenwoordig zijn upgrades gratis voor zowel IOS als Windows (vanaf Windows 10).