[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to the OWASP Top 10 Web Application Security Risks for ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=0) [I'd like to start with a brief introduction of what we're going to cover in this course and the first thing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=9) [I want to talk about is who is getting hacked and I normally like to start here just to give a sense of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=15) [breadth of websites which are coming under attack from online hackers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=21) [I'd also like to talk about who is doing the hacking and those attackers frequently break down into the same](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=26) [three groups that we'll take a look at. We'll then move on to having a look at what OWASP is and where this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=32) [Top 10 has come from and how it's going to apply to this course. And finally, before we start getting into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=38) [those Top 10 risks, I want to talk briefly about applying security in depth because that's a very important](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=45) [theme that's going to keep reoccurring throughout this course.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=0&mode=live&start=51)

[Who's getting hacked?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live)

[Usually when I start out with a presentation about security, I like to show this slide and this, unfortunately,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=0) [is a slide of many, many different organizations who have been hacked in recent years.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=7) [Now the reason I show this is it gives a good understanding of the breadth of websites which are coming under attack.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=13) [Many of these logos will probably be familiar to you, but let's touch on a few of these to get a bit of a sense](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=20) [of the variety of websites. So, for example, Sony we know had a terrible year in 2011 with many, many attacks,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=26) [Obviously a multinational electronics producer. Then we go to someone like Lockheed Martin.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=34) [These guys make military jets. This is a very, very serious organization.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=39) [Then at the end of the scale, we have somewhere like LUSH, who basically make bath produces and soap and so a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=44) [very, very different target to someone like a military contractor.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=50) [Then we go all the way through to more social orientated sites like the PlentyofFish dating site and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=54) [all the way through to sites like pron.com for adult entertainment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=60) [So, there's a huge variety of different organizations here. Nobody is beyond the reach of determined attackers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=63) [and that's the first important message I'd like to get across in this course.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=71) [One figure I'd like to look at before we get started to try and give a sense of the prevalence of web security](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=77) [risks is this particular statistic from the website security 101 report that WhiteHat Security produced in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=82) [May 2012 and what they found from a huge range of analysis of websites was that 90% of those websites have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=90) [serious security flaws. Now, of course, there are many different flaws of many different natures, but that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=98) [9/10 figure is absolutely staggering and hopefully gives you a bit of a sense of how prevalent this risk is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=104) [and why courses like this are so important.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=1&mode=live&start=112)

[Who's doing the hacking?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live)

[Before we get into this course, I'd like to talk a little bit about who is doing the hacking because understanding](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=0) [where these threats are coming from is extremely important in order to help us protect our website assets.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=5) [Now the thing about online attackers is that they do have very, very different roles and very, very different](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=13) [motivations. Different groups of attackers also have access to very different degrees of resources.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=19) [Some of them have very few resources at their disposal, some of them have almost limitless resources, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=26) [depending on the nature of your site, you're potentially trying to protect your assets from two very different groups.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=31) [Now, of course, when we are protecting website assets, we're never really getting to a position of absolute security.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=39) [None of what we're going to talk about in this course is about making a site unhackable.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=46) [That's simply not a phaseable position to be in against an attacker with enough time and enough money.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=52) [Instead, we're always trying to find the right balance between how much we invest to protect what the value](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=59) [of the asset we're protecting is, and what the likelihood of an attacker exploiting it, and of course,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=65) [the impact if an exploit does occur. So, I'd like to think it's a very pragmatic approach.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=70) [It's about being sensible about how we apply security, not underdoing it, but not going overboard.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=76)[We need to find the right balance and that's always going to differ site by site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=82) [So, let's consider that balance in the light of these three different categories of attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=87) [The first category is probably very well known. They are the Hacktivists.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=94) [So, these are the guys who often claim to be motivated by some greater cause.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=98) [They're protecting the downtrodden or overthrowing regimes or providing some other service that they see as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=102) [a social justice. Unfortunately in many cases, it turns out to be a much more opportunistic group.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=109) [We've seen many occurrences of them attacking not-for-profits or other organizations that really don't fit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=116) [the hacktivist creed of that greater cause. We often find that these individuals are very young,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=122) [frequently only teenagers or early 20s. Consequently, they're less experienced and they're also less aware of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=128) [the social consequences, which of course in itself makes them a little bit dangerous because they possibly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=133) [don't always have the forethought about what the outcome might be if they do attack a website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=138)[Fortunately, though, they also don't tend to have a lot of funding so they're not a particularly well-resourced](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=144) [group, certainly not compared to the next two that we'll look at, and of course as many of you have probably seen,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=149) [after an attack they do tend to be quite vocal, they like to talk about it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=155) [Now we know these hacker collectives by many different names. A couple of the more famous ones, though,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=158) [would be the likes of Anonymous and LulzSec.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=164)[The next group is online criminals and these guys are predominantly motivated by cash.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=168) [They're attacking websites for financial gain. Now the kind of information that they're going to be interested in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=174) [is things like financial data. So, could they get, say, credit card details from a website and then either](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=179)[use those directly or sell them on an underground market. There is an absolutely thriving black market for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=185) [this sort of asset with credit cards sold in highly-organized marketplaces in bulk via online criminals to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=191) [other online criminals. There's a clear financial motivation. Likewise, with personal data, so personal data is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=198) [regularly used for identity theft, which of course poses all sorts of financial opportunity to career criminals.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=204) [And, again, often that's sold by our underground black markets because there is a monetary value to it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=211) [Criminals are also looking at ways of distributing more ware or creating botnets because once they can do that,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=217) [they can then sell the services of those botnets. Botnets give criminals the ability to do anything from launch](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=222)[denial-of-service attacks to distributing spam. So, again, there's a monetary value to it and that attracts](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=229) [the criminals. Now as criminals, they really are running some form of a business. They do have some degree of funding,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=235) [they are looking for an ROI on their investment, they want to attack sites that have something of financial](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=241) [gain to them, and when necessary, they will invest in order to make a return.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=247) [So, they can be quite organized and run themselves like a business would.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=252) [The third category is the nation states. So, these are countries that are engaging in what we're commonly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=258) [referring to now as "cyber warfare". Now nation states are predominantly going to be interested in assets of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=265) [national security or political interest. So, these activities may occur against another country who they feel](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=272) [is a threat, but they may also be domestic. We've seen many examples in recent years of governments using](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=277) [technology and using hacking techniques against websites in order to target their own citizens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=283) [Nation states are extremely well funded. Of course, it differs country by country, but there can be an almost](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=289) [unlimited pool of money available and when there's enough money and enough smart people, there's pretty much](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=296) [no target that is beyond their reach. How you would protect a website of interest to a nation state and the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=302)[amount that you would invest to do so is going to be fundamentally different to the way you protect it from](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=309) [a hacktivist and that's really the point of looking at these three different groups -- to understand the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=316) [different risk profiles and the investment that is required in order to properly protect the asset.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=320) [When you think about the sort of investments that nation states have been able to make, think about initiatives](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=326) [like Stuxnet and Duqu and Flame. So, in the case of something like Stuxnet, we're talking about cyber warfare](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=332) [against uranium-enrichment centrifuges, airlocked off the network with more ware distributed by a USB going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=340) [undetected for years and having enormously complex implementations of code in order to attack those assets.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=347)[That is a very serious investment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=2&mode=live&start=355)

[OWASP and the Top 10](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live)

[Let's move on to OWASP and the Top 10 and the first thing is, is that OWASP is an acronym and it stands for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=0.5) [the Open Web Application Security Project. OWASP is a not-for-profit worldwide charitable organization and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=7.5) [their focus is around improving security for websites. Now OWASP does a number of different things and creates](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=15.5) [a number of different publications and the one that we're going to be looking at in this course is called](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=22.5) [The Top Ten Most Critical Web Application Security Risks and the great thing about the Top 10 is that it is a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=27.5) [technology agnostic guide for the way you secure your web applications.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=34.5) [By technology agnostic, I mean that we can apply that equally to ASP.NET, as we could to PHP or Java or any](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=40.5) [other web-based technology. OWASP talks very much about HTTP, HTML, and other risks that are common across](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=49.5) [the various service decks. Whilst that broad-cutting objective is in many ways very good, it also makes it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=58.5) [difficult to implement it in one specific technology stack and that's why we have this course here so that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=65.5) [can look at precisely how we implement this in the ASP.NET stack because simply, the way we mitigate a risk](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=72.5) [at the actual server level is going to be very different in ASP.NET to what it would in other server-side](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=79.5) [technologies. One of the things that makes the Top 10 such a valuable resource is that it is very frequently](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=85.5) [a reference point both for security specialists and for developers. So, for example, it is very common for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=93.5) [penetration tests or website scans to come back and reference items within the Top 10 when risks are found.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=100.5) [So, the Top 10 is almost like a conical reference for web application security.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=108.5) [It also very frequently forms the security guidelines for many web developers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=113.5) [So, a lot of developers start out with the Top 10 in mind. When they're building features of the web application,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=118.5) [they're making sure that each one of those items in the Top 10 is sufficiently mitigated in the development work](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=125.5) [they're doing and it's the ubiquity of the Top 10 that makes it so worthwhile learning.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=132.5) [The 10 most common web application security risks are ranked in order of prevalence and by far the most prevalence](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=139.5) [is Injection, which we commonly think of as SQL injection, but it is a little bit broader than that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=146.5) [Next up is Cross-Site Scripting or XSS and then we'll move on to Broken Authentication and Session Management,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=151.5) [which covers a variety of different aspects of how we persist state in this stateless protocol that is HTTP.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=158.5) [We'll then move on to #4 about Insecure Direct Object References and look at how we need to protect individual](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=166.5) [records within our database and then on to Cross-Site Request Forgery, which is all about an attacker tricking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=172.5) [a browser into making requests on their behalf. We'll move on to part 6 on Security Misconfiguration,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=179.5) [which covers quite a broad range of different aspects of the way we secure our application and then on to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=186.5) [Insecure Cryptographic Storage and we're going to spend a lot of time talking about password storage,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=193.5) [as well as looking at some symmetric encryption. Part 8 will take us on to Failure to Restrict URL Access and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=198.5)[we're going to look a lot at authorization and protecting resources and then we'll move on to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=205.5) [Insufficient Transport Layer Protection, so we'll be talking a lot about HTTPS, SSL, and TLS, and particularly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=210.5) [how unencrypted traffic on the network is at risk, and then we'll wrap the whole course up by looking at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=218.5) [Unvalidated Redirects and Forwards, which many of you are probably familiar with, but may not be fully aware](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=224.5) [of the potential risk that they pose.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=230.5)[As we move through the Top 10 for each of those top 10 risks, we're going to refer back to a common matrix.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=234.5) [This matrix will begin by looking at the Threat Agents, so who might the attackers be for this particular risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=240.5) [We'll then look at a variety of different Attack Vectors, so how might the attacker actually exploit a risk,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=249.5) [how will they get into the application. We'll then talk about the specific Security Weaknesses that that attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=255.5) [can expose and often we'll find that there might be multiple weaknesses for the one risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=262.5) [We'll then move on to Controls and the important thing about controls is that we are going to try and apply](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=268.5) [multiple controls to each risk and we'll talk about that more in just a moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=273.5) [The Technical Impact is going to be very dependent on the asset that is being protected.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=279.5) [So, we'll talk about that in the different context of each risk as we progress through the course.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=284.5) [And finally, the Business Impacts and, of course, that is also going to be very specific to the nature of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=290.5) [data that the particular application is protecting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=296.5)[When we look at this matrix for each risk, we're going to assess each one of these items against a basic](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=301.5) [traffic light system just to try and give us a good visual idea of the relative risk against the other risks](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=306.5) [in the Top 10 and this is a grading system provided by OWASP.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=312.5) [So as we progress through the course, we're going to see this same matrix applied to each of the Top 10 risks,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=316.5) [we'll spend a couple of minutes having a look at it in context, and then we'll move on to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=321.5) [actually understanding and mitigating the risk in detail.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=3&mode=live&start=325.5)

[Applying security in depth](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live)

[As we move through the Top 10 in this course, we're going to be applying what is referred to as security in depth.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=0) [Well you'll often also see it referred to as defense in depth and the sentiment here is that when we try and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=7) [protect against a risk, it shouldn't just be one defense that we put in place.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=12) [We want to lay these defenses; we want to try and build up multiple barriers so that there is no single point of failure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=17) [So as an example, we're going to be looking at cryptographic storage of passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=25) [Now, of course, this is really only of much value if an attacker gains access to the passwords in storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=29)[Now up stream of there, we're going to look at multiple defenses to try and keep an attacker out of our storage](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=35) [to begin with. So, in this case, cryptography is a last line of defense, but it is that deep defense.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=41) [It's several layers under the other defenses and that is an extremely important theme we're going to keep](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=47) [coming back to in this course.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=53) [The other thing that's important when we talk about the depth of security is that it doesn't end with this Top 10.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=57) [The Top 10 is an excellent start. Indeed, I think it's the best possible start that we can give developers and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=63) [as we go through that Top 10, we're going to develop a deep understanding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=69) [So, we're going to go through the definition of the risk, we're going to exploit each risk so you will see an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=73)[ASP.NET application hacked 10 different times using those 10 different risks, and then we're going to apply](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=79) [multiple ways of mitigating the risk. What we need to remember is that security does go beyond just the technology.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=86) [So, there is always the human aspect of security, what are the business processes around the way applications](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=94) [are managed, and what possible risks do they introduce? Then, of course, there's always social engineering.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=99) [So, even if we build secure systems, is it possible to circumvent those controls by engineering vulnerable people](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=105) [and ultimately we are all vulnerable to some degree of social engineering.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=112) [Plus, of course, there are other risks that fall outside the scope of the Top 10.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=115) [These are going to be the ten most common risks that we look at, but they're not the only risks.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=120) [And finally, we need to remember that this is a rapidly evolving landscape.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=125) [Indeed, technology itself is a rapidly evolving landscape so it only serves to reason that attackers are going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=129) [to continually invent new ways of attacking applications and certainly there is an arms race that happens](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=135) [between attackers and the people building their systems. Two-factor authentication is a great example.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=141) [It's a fantastic security measure, but attackers are already working out ways to circumvent that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=147) [Then, of course, there are always new risks and attack vectors emerging.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=152) [Partly that's because the technology's evolving and partly it's because the attackers are evolving.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=156) [Either way, it means that the risk profile continues to change.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=161) [So, ultimately, apply the Top 10, learn it in depth - it's an excellent resource, but even when we do a great job](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=166) [of the Top 10, it is still absolutely essential that we stay vigilant.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m0-introduction&clip=4&mode=live&start=172)

[Injection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to the first module of the OWASP Top 10 on Injection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=0) [In this module, we're going to start out by having a look at how OWASP views the injection risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=6) [We'll then move on and perform an attack against a vulnerable application so you will actually see multiple](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=12) [ways of exploiting SQL injection in an application. We'll then take a closer look at SQL injection and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=19) [understand the relationship with untrusted data. From there, we'll drill down into the database and look at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=25) [how we can implement the principle of least privilege in order to restrict the access rights that the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=32)[web application has to the database. We'll also take a look at parameterization in inline SQL, as well as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=37) [how we can use stored procedures to reduce injection risk and we'll also have a look at how stored procedures](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=45) [can actually still be vulnerable to injection as well. We'll then move on to take a look at whitelisting](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=51) [and how we can validate our untrusted data against a whitelist using one of several different mechanisms.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=56) [We'll also take a look at Object Relational Mappers and how tools such as Entity Framework can protect](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=63) [against SQL injection through their use of parameterization. And finally, we'll have a look at some remaining](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=68) [risks, as well as how an attack can be really easily automated using freely available tools.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=0&mode=live&start=74)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live)

[As we look at each of the OWASP Top 10, we're going to begin by referring to a small matrix, which helps us](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=0) [get a better understanding of the risk. This matrix is going to help us understand the Threat Agents,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=6) [so where a text might come from. It's also going to look at the Attack Vectors, which is how that Threat Agent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=12) [may actually attack the application. We'll then look at the specific Security Weakness targets.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=18) [What is it about the application that makes this risk possible? It also addresses the Security Controls that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=24) [we might put in place to mitigate that risk. Then we'll take a look at the Technical Impact.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=30) [So, if this risk is successfully exploited, what might that mean to the application?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=36) [And finally, we'll look at the Business Impact. So, if the risk is actually exploited, what might it mean](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=40) [to the business, which is running the application? As we look at each one of these items in the matrix,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=46) [we'll also see them graded. The grading uses a basic traffic light system, which will give us a bit of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=51) [an indication of where we need to focus our effort when trying to mitigate the risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=56) [So, let's start by looking at the overview and risk rating for injection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=63) [Now the first thing we'll say about injection is that the Threat Agents can be anyone who sends untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=67) [to the system. We'll talk about untrusted data more in this module.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=73) [For now, let's assume that could be anyone who's an external/internal administrator, basically anyone who has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=79) [access to the system is a possible threat. The Exploitability is extremely easy because it is ultimately](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=84) [just a simple text-based attack and we'll see how easy that can be as we progress through this module.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=91) [In terms of Security Weaknesses, the prevalence is pretty common and the detectability is average.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=96) [Another thing to point out just here is that when we talk about injection, it's not necessarily SQL injection,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=102) [which is the term that we normally associate it with. It could be other forms of query injections, such as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=109)[LDAP and XPath. The impact of injection is severe. So I consider something like SQL injection where the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=115) [result may be that internal data is modified or disclosed and we'll see just how that happens in just a moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=122) [The Business Impact will depend on the nature of the data and, of course, on the nature of the business, but we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=130) [need to assume that it could be serious given the ability to disclose internal information to unintended recipients.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=1&mode=live&start=134)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live)

[I have an application running on a local instance of IIS on my machine and it has a host entry to map the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=0) [address injection through to that IIS instance. This gives us a pretty good reflective representation of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=7) [what we might see in a normally externally-facing website. Now the front page of this website has a number](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=14) [of product categories listed on them and these categories come out of the AdventureWorks database,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=21) [of which I'm using a trimmed down version for this demonstration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=27) [So, let's drill down and take a look at the Mountain Bikes category and what we'll notice when we look at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=31)[Mountain Bikes is a fairly typical pattern, which is that our URL goes to our Product Page and it passes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=37) [through a subcategory ID of 1. Now it's easy to speculate that if we were to change that ID, we would get](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=44) [a different series of products. So, if we make that, say, 2, we now seem to get what appear to be road bikes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=53) [So, clearly manipulating the query string parameter is changing the database query.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=62) [So, extending that logic a little bit further, let's start to manipulate the URL parameter in the query string](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=67) [and see if that changes the data that is being returned and a very typical pattern that people test for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=73) [with SQL injection is to put an or condition in the query. Now we have 43 products at the moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=79) [Let's now change that query and you'll see that we now have 504 products and what this suggests is that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=87) [somehow we are actually able to change the filtering. So, perhaps that SQL query is no longer just filtering](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=95) [to product category 2, which was our road bikes; clearly we're adding a condition here, which is returning](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=101) [a significantly larger number of records. Now so far we haven't actually retrieved any data that is of a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=107) [sensitive nature or performed any malicious tasks. We've simply changed the execution of the query to give us](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=114) [more data than what we would normally expect. Let's now go through and change that query string parameter](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=120) [to a string value. Now what we see here is that first of all custom errors haven't been correctly configured](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=127) [and we're going to talk more about that later on in the later module about security misconfiguration,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=134) [but this is actually going to be very helpful for exploiting SQL injection because we're now starting to see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=140) [errors come up from the SQL Server, we are seeing SqlExceptions, and it's giving us information about the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=146) [internal implementation of the database, such as the fact that x is an invalid column name.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=153) [So, it's starting to tell us a little bit more about the query structure and it also tells us that we can get](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=159) [errors generated by the SQL Server and retrieve them in the browser, which is going to be very, very handy](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=164) [for trying to exploit this SQL injection risk a bit further. So, let's now take a look at how we might](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=171) [actually get data out of the system, which is not intended to be exposed publicly.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=176) [So, what I would like to do is go and copy and paste a query, which I've prepared in advance, and this query](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=182) [is going to start to disclose internal structure in terms of tables that might exist.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=189) [Now what this query has actually done is it's disclosed the name of an internal table in the database and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=197) [it's done that by creating a very carefully crafted query that we can now manipulate to give us even more tables](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=203) [from the database. So, we'll have a look at what the actual SQL query looks like with syntax highlighting](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=211) [in a moment, but for now, if we start to change values, we'll see that there's a ‘ProductSubCategory' table](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=216) [and if we go back and increment that by another number, we will also see that there is a ‘CreditCard' table](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=227) [in the database. So, let's go and take a look at what this looks like with some syntax highlighting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=235) [So, here's the last query that we just ran by injecting it into the query string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=241) [I've put it in SQL Server Management Studio so we have syntax highlighting and I've put in some line returns](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=246) [and some spaces to make it a little bit more legible and as you'll see, it's actually three nested select statements.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=252) [So, the first one begins by taking the top 3 records and 3 was the last variable that we manipulated in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=259) [query string. It takes the top 3 id's from sysobjects, which lists all of the objects in the particular](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=266)[database that we're querying. It gets the ones where the xtype is char(#5).](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=274) [Now xtype is the object type and char(#5) is the S key escape character for u or user table and the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=279) [escape character is being used to avoid any problems with passing a quoted string through in the query string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=288) [It's then ordered by the id and it is ordered by ascending. That then nests into a statement, which just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=296) [takes the first one and it is ordered by descending. So, by selecting the top 3, we're then saying take the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=302) [last one and this is the way we could keep incrementing it because when we took the top 2 and we took the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=309) [last one, we go to different value. So, you can enumerate all the way through until you run out of records.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=314) [Now that id is then used to select the name out of sysobjects and it's that name, which then returns us the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=320) [table name, such as CreditCard that we saw in the last example. Now the reason this actually works is because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=327) [when this query runs, we will have a string value. The reason it was exposed in the web interface is that that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=334) [string value then gets compared to the id column, which is an integer which is why we got the exception we saw earlier.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=342) [So, if we go back and look at that exception, we can see that it's a conversion failure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=351) [So, we got the CreditCard string value and we tried to compare it to an integer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=355) [Now the reason why we do this is that because custom errors are off on this website, if we can cause the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=361) [website to throw an error like this and bubble up from the SQL Server and land in the web interface,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=367) [we can start to retrieve internal information about the system, so now we have a table name.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=373) [Let's take that logic a little bit further. Now that we have a table name, we can start to construct queries](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=380) [to give us things like column names and once we have column names, we can construct queries to actually select](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=386) [data out of there. So without going into the data of what's actually inside the query, here is a query](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=393)[which will start to expose credit card numbers from the database and here we go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=399) [Exactly the same process in terms of actually getting a piece of data, causing an internal SqlException, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=405) [then allowing it to bubble up into the user interface. So this has been a very successful SQL injection attempt.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=411)[Now this example worked very well because the internal SqlExceptions bubbled up into the web interface,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=418) [which allowed us to retrieve data from the SQL Server. Even when we can't do this, it's still possible to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=425) [execute SQL injection through what's known as blind SQL injection and there are other ways that blind SQL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=432) [injection can still get data out even if it can't actually show it in an exception message like you see here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=438) [So in short, the problem here is not just the fact that we have custom errors incorrectly configured and,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=446) [again, we'll look at that in more detail when we get to security misconfiguration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=451) [There are other vulnerabilities in this application that actually allowed the SQL injection to be executed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=455) [in the first place.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=2&mode=live&start=461)

[Risk in practice: LulzSec and Sony](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live)

[Now to try and give a sense of just how severe SQL injection can be, let's take a look at an example that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live&start=0) [happened back in 2011 to Sony Pictures. Now what happened in this case was a Hacktivist Group by the name](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live&start=6) [of LulzSec was able to compromise sonypictures.com and compromise over 1 million users' personal details,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live&start=13) [which included things like passwords, email addresses, and home addresses.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live&start=23) [LulzSec also claims that this was all possible by very simple SQL injection and, in fact, after Sony Pictures,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live&start=28) [we saw numerous other exploits of Sony, not just Sony Pictures, but other Sony websites occur with absolutely](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live&start=35) [staggering costs. SQL injection was often the attack vector because the impact is so severe.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live&start=43) [The leaking of personal data was a huge problem for Sony and it had a massive impact on their business.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=3&mode=live&start=51)

[Understanding SQL injection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live)

[Now that we've seen SQL injection in action and we've seen a real-world example by Sony Pictures as well,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=0.5) [let's start to understand what makes SQL injection possible. So, let's take a URL such as we see here -](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=6.5) [mysite.com/Widget?Id=1. Now what frequently happens is a URL such as this translates pretty directly through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=14.5) [to a SQL statement such as this. Now all of this probably looks pretty obvious, but what I'd like to try and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=25.5) [explain a little bit better is that there are really two parts to both of these lines that you see here and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=32.5) [the first part is that everything on the left-hand side is what we would refer to as trusted.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=38.5) [So, for example, when we look at the URL, this is a page which we control, it is part of the application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=44.5) [Likewise, when we look at the SQL query, this is a query that we control.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=52.5) [We, as developers, have written this somewhere in the application, we know this to be safe.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=56.5) [On the other hand, the data, which is appended to both the query string and then ultimately the SQL query,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=62.5) [is what we would refer to as untrusted and there's a very important difference](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=69.5) [because the untrusted data is where the attack comes from.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=4&mode=live&start=74.5)

[Defining untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live)

[So, let's take a look at what we mean by untrusted data and the first thing is, is that the integrity of it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=0) [is not verifiable so we have no idea if it is legitimate. In the earlier example, we were meant to land on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=5) [the product page by following a link. So, the query string would naturally be something that is already](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=13) [coded into the application, it was just in the hyperlink on the front page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=18) [We managed to compromise the integrity of that query string by manually manipulating it in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=22) [Now that then brings us onto the second point about the intent possibly being malicious.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=29) [So, in that example, we were able to actually change it in order to do something malicious to the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=33) [So, we managed to discover tables, columns, and then ultimately extract data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=39) [So, we must assume that untrusted data could be malicious in intent. Now this is not just an injection risk](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=44) [as well and we'll talk about untrusted data a lot in subsequent modules.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=52) [Untrusted data may include payloads such as SQL injection, which we saw just before, cross site scripting,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=57) [which we'll see in the next module or even binaries containing malware.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=64) [If there's an upload function for a website, for example, that is untrusted data, we can't verify the integrity,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=68) [and we must expect that the intention of the data may be malicious.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=75) [So, let's now take a look at some of the places that untrusted data can come from and there are a few common places.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=80) [So, the first one is from the user and in our example earlier on, the URL contained untrusted data via the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=87) [query string parameter. The user could also submit untrusted data via a form.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=93) [So, we would need to expect that any data in a post request could have malicious intent.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=98) [The other source that untrusted data could come from is automatically from the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=104) [So, for example, the browser automatically sends any cookies for a domain when it makes a request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=109) [Those cookies could be manipulated by an attacker. We can't assume that just because we get a cookie for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=115) [the domain, which we hope that we set originally, that may not be the case; it may actually be malicious.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=121) [It's the same thing for request headers. So, every time the browser makes an HTTP request to a website,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=128) [they are request headers that contain things such as the user agent, which is the browser and the operating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=134) [system that the user has, as well as things like the language preference of the user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=139) [They should be automatically set in the browser, but they can be manipulated by an attacker and they could](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=145) [contain a malicious payload, but there are other locations that untrusted data comes from as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=150) [So, for example, external services. If an application consumes a web service from an external source,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=156) [we would need to expect that that data is untrusted and could be malicious in intent.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=163) [And finally, your own database could contain untrusted data. If there is a feature for users to submit data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=168) [to a website and for that data to then be saved into a database, it contains untrusted data and we would need](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=174) [to assume that every time we handle that data in the database, it could be malicious in intent.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=5&mode=live&start=181.5)

[Demo: The principle of least privilege](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live)

[So let's now start to look at how we're going to protect this application from the risk of SQL injection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=0) [Now with most security mitigations, we want to apply what's referred to as security in depth and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=7) [what that means is that we don't want to apply just one mitigation; we want to try and protect in layers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=12) [When we do this, if any one layer has a vulnerability, we've got to back out, we've got an additional protection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=19) [So, this security in depth is extremely important and it's a topic that will come up a lot through this course.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=25) [In the case of SQL injection, what we're going to do is start protecting the application from the database](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=31) [all the way back up to the web tier and we're going to apply these mitigations one by one by one and see how](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=37) [each one protects the application further. Now where I'd like to start protecting this application is by](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=45) [applying the principle of least privilege. Now what this principle says is that every module in a computer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=52) [system must be able to access only the information and the resources that are necessary for its legitimate purpose.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=58) [In other words, make sure that every account can only do the absolute bare minimum that it needs to do.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=66) [Now this is important because in the earlier example we saw how an anonymous user of a website was able to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=72) [retrieve credit card details. This is not consistent with the principle of least privilege.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=78) [An anonymous user should not be able to just come along to a website and start to pull credit card details out.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=84) [This is not something that we should be exposing externally. So, what we need to do is make sure that when](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=91) [that web application connects to the database, it can access things like products, which it needs,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=97) [but it can't access credit cards.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=103) [So let's go and take a look at SQL Server and have a look at our database and in this case we're using the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=106) [injection database and I would like to go through and have a look at the permissions on the one injection user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=111) [Now this is the account that the web application is connecting to and the first thing that we'll see here in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=118) [Membership is that it's a member of db\_owner. Now db\_owner is way too excessive.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=124) [db\_owner will grant access to read, write, update, delete, any table at all in the database, and that's excessive.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=130) [It will also allow the dropping of objects and the creation of objects, which opens up all new other risks.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=139) [So, we definitely do not want to grant db\_owner. So, if we want to apply the principle of least privilege,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=144) [what we really want to do is go through and give the user access just to the tables that they need.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=150)[So, let's jump into our Securables and what we're going to do is go and search for all objects of types and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=156) [what we want here is Tables and when we get those back, what we want to do is give that user select access](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=164) [to the product table and we'll want to give them select access to the ProductSubCategory table.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=171) [They don't need access to anything else and we've already removed their access from DBO as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=180) [So let's just say OK to that. Okay, now we're back at the web application so the first thing we'll do is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=186) [reload this page and make sure the user still has access and that still works so that's great.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=191) [Now let's go through and we'll choose Mountain Bikes again and that still works because the user has been](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=196) [explicitly granted select access to the Products table. Now let's go back and grab that malicious query,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=202) [which tried to pull data out of the CreditCard table and let's see what happens this time when we try to run](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=210) [that query and here we go -- SELECT permission denied. So, the user could not get any data out of the table](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=216) [because they simply no longer have access. So, the principle of least permission is a very, very easy access](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=222) [control, which immediately stops the ability to get access to data that the user should never have the ability](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=230) [to select from. Now what we need to remember with the principle of least privilege is that we may have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=236) [different roles in the system. So, whilst our end user may not have needed access to that data, maybe an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=242) [administrator does. This may result in having multiple SQL accounts in an application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=248) [Whilst there may be some additional work in doing that, it does significantly increase the security position](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=254) [because it means that anonymous users can simply never get access to that data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=261) [It's a very, very secure position to be in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=6&mode=live&start=265)

[Demo: Inline SQL parameterisation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live)

[At the heart of most SQL injection vulnerabilities is simple](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=0) [string concatenation between query and untrusted data and we can see this happening down here where we have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=4) [a query and then we have untrusted data by virtue of the ProductSubCategoryID, coming from the query string,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=10) [and simply being joined up into one great big query where it's then executed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=19) [This is what allowed us to manipulate this value and actually change the function of the query.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=23) [The real problem with this is that the query is not parameterized so what we're going to do now is turn this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=30) [into a parameterized, inline, SQL statement. Now in order to do this what we're going to do is no longer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=37) [just concatenate the data to the query. We're actually going to declare this as a parameter in the query like so.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=45) [Now once we've done that, we can simply pass that parameter as a property of the command object.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=54) [So, what we're going to do is go back to the command object and we're going to look at the Parameters collection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=61) [and we're going to add to that Parameters collection and the overload we're going to use is an overload that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=67) [passes a parameter name and then it passes a SqlDb type. Now our parameter name is simply what we have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=74) [up here in our statement and we'll pass that through as a string and then the type we're going to use for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=83) [this example is going to be a VarChar type. Now what we're going to do is set the value of that Parameter](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=90) [to the value that we have read in from the query string like so. So, what this now means is that we're going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=99) [to pass the same data; we're just simply passing it in a parameterized fashion.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=108) [So, we're not concatenating the string, it is in there as a SQL Parameter, and we're adding that parameter](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=113) [to the command object. So, let's rebuild and then we'll go and check how this looks in Internet Explorer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=119) [Okay, so we're back on the Product page, let's now refresh that page and see what we get.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=126) [So, the page loaded fine, which is what we'd expect because this is a normal legitimate request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=134) [Now let's try and put in that original or statement and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=139) [So, what's happened is we've still got a SqlException so it has gone all the way through to the database,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=145) [but what's happened this time is the conversion has failed because what it's trying to do is take that entire](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=149) [parameter, which is obviously not an integer value, and it's trying to compare it to a column type of int,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=155) [which is the ID. Now the reason why this has happened is because it's just taken this entire query string](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=162) [value and it's tried to do an equivalency test to a column that is an integer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=169) [So it hasn't broken out that query context anymore; this is now a separate individual parameter and that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=174) [is what'll stop the SQL injection from happening.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=7&mode=live&start=181)

[Demo: Stored procedure parameterisation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live)

[Another very robust way of mitigating the risk of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=0) [SQL injection is to use stored procedures. In reality, string concatenation of the query, such as we see here,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=3) [is really not ideal, even when it's parameterized. There are many reasons why this is not a good design pattern.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=11) [So let's now replace this with a stored procedure and what we'll see is that the stored procedure will offer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=19) [the same mitigation against SQL injection. So, I'm going to replace the existing inline SQL statement here](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=25) [with a query I created earlier called GetProducts. So, that's the name of our SQL stored procedure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=31) [Now I'm going to replace the command type to refer to a stored procedure, CommandType.StoredProcedure,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=39) [that is it. Let's now rebuild that and we'll go back and run it again in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=49) [Okay, we're back in the browser. Let's reload a normal legitimate request again and that one works just fine](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=55) [and now let's try our SQL injection again and it doesn't work and this time we have a slightly different](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=62) [error message so it's another type conversion error message so it has had problems converting to an integer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=68) [So let's take a look at the stored procedure and see why that is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=75) [So, I'm back in SQL Server and here is our stored procedure so let's have a look at what that guy looks like](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=79) [and this is a fairly typical sort of procedure and what we see here is that the parameter is expected to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=86) [an integer, so this has failed. Now even if it was expected to be a string value, we still wouldn't be able](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=94) [to get any injection through because it's parameterized. So, when we actually put that parameter here in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=102) [query, we can't break out of this data context and actually enter the query context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=108) [So, whatever the value is that's in that variable, we won't be able to break out and actually into that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=114) [query context and change the execution of the stored procedure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=8&mode=live&start=119)

[Demo: Whitelisting untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live)

[So, what we've done with the mitigation so far is gradually protected layers of the application from the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=0.5) [database back to the web tier and what I'd like to do now is actually move the mitigation even further up](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=5.5) [the request pipeline by implementing some whitelisting. Now in theory, what we want to do is make sure that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=12.5) [all untrusted data going into any system is validated against a whitelist of known good values.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=18.5) [Now what this means is that we want to be very, very explicit about the data that we allow to enter into a system](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=26.5) [and a whitelist is a very explicit control because what a whitelist does is it says, these are the characters](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=34.5) [or the data values which we are going to allow to enter our system for this particular parameter.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=41.5) [Now that's a very strict filter because it will filter out anything that we don't explicitly trust.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=48.5) [The alternative is a blacklist. Now blacklists can be dangerous because a blacklist is sort of the inverse](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=54.5) [of a whitelist, so a blacklist says these are the things that we know are bad, that we don't trust.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=62.5) [Now that's rather implicit because it implies that if we don't know something is bad, it must be good,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=69.5) [so let's allow it. The problem with that is that there will inevitably be things that may be bad later on,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=75.5) [but we just don't know about it now. One thing about security is that it does continually evolve so we see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=82.5) [new attack vectors for things like SQL injection and cross site scripting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=89.5) [We can't just rely on everything we know now still being correct in the future.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=93.5) [This is why the whitelist works so well because it simply says, what is the data that we need right now,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=98.5) [what do we trust, what are we happy to allow. Anything else shouldn't be allowed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=104.5) [So let's talk about how we might implement a whitelist and there are a few different approaches that we can take.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=112.5) [Now the first approach and this is the one that we'll implement in a moment is around type conversion.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=117.5) [So, for example, if we can convert a piece of untrusted data to an integer, it is highly unlikely that that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=123.5) [untrusted data will be able to contain a malicious payload. Certainly it won't be able to contain things like](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=131.5) [spaces and equal signs and quotes, which we often see in an attack.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=137.5) [Type conversion might also be the type such as dates or global unique identifiers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=143.5) [If we can get away from strings, we do mitigate a significant amount of risk because strings are often the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=148.5) [attack vector. When we do need to use strings, using regular expressions to whitelist the allowable values](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=153.5) [is a very, very good control. So, for example, if we were to accept an email address, we know that an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=161.5) [email address must comply to a particular pattern. There's a spec that we can implement and we can put](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=168.5) [into a regular expression. It's going to be much harder to get a malicious payload through a regular expression](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=175.5) [for an email address than it what it will be for a string that has no whitelist on it whatsoever.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=181.5) [We could also write regular expressions for things like phone numbers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=188.5) [Phone numbers often have spaces and brackets and dashes, but don't normally have equal signs or quotes,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=191.5) [so you wouldn't allow that within a phone number, but again, we're not going to be explicit about what we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=198.5)[don't allow (that would be a blacklist), we're going to be explicit about what we do allow, which is a whitelist.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=203.5) [We could also implement regular expressions around names or other string values, but we do need to be very careful.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=210.5) [Often when we talk about security, there is a balance between usability and the security of the control that's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=218.5) [being implemented. We don't want to tighten the security to the extent where legitimate user input doesn't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=225.5) [get through. There have been many precedents of people writing controls and applications that don't allow](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=232.5) [single quotes to be entered into a name. Now this doesn't make much sense because, of course, there are a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=238.5) [huge number of names out there that do legitimately contain a single quote, particularly when you start to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=243.5) [look at names from other countries that may use non-roman alphabets.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=249.5) [There are a lot of characters, which are legitimate characters and are a legitimate part of someone's name.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=252.5) [You don't want to exclude those guys from your system because you've created an overprotective regular expression.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=258.5) [And finally, another approach to whitelisting is simply to have a list of known good values.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=265.5) [So, these are the string values that we will accept. So, for example, let's imagine somebody was choosing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=270.5) [a country from a list and for the sake of this example, let's assume that rather than an ID, it was actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=277.5) [passing the string value of the country, the name of the country.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=283.5) [It would be very easy to say that if somebody tries to submit a country that doesn't already exist in the system,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=286.5) [then that request should be rejected because it wouldn't pass the whitelist validation; it wouldn't be one of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=292.5) [the known good values. So, let's go back to our application and look at implementing a whitelist validation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=298.5) [for the ProductSubCategoryID that comes in via the query string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=306.5) [Now the obvious whitelist validation approach for this is that we're going to do a type conversion,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=311.5) [so our identifiers or integers. So, this actually makes it very easy so let's just start out by declaring](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=316.5) [a new integer and we'll call it id and what we want to do now is actually do a TryParse on this.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=322.5) [So let's go int.TryParse and the value that comes in will be our ProductSubCategoryID from the query string](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=329.5) [and the value that comes out will be the variable that we just created.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=337.5) [Now, if this does not pass TryParse, then our input is not an integer and it does not pass the whitelist](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=342.5) [validation so let's just throw a new ApplicationException and you would probably handle this more gracefully](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=350.5)[at other times, but this will do for this example -- ("ID wasn't an integer"). Okay.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=358.5) [Of course the other thing we can now do is that we've actually got our untrusted data as an integer in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=366.5) [this id value. Let's actually go down and when we do pass this to our StoredProcedure, we can pass that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=371.5) [as an int and we can change the string value back here and actually pass it as an ID.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=377.5) [So let's rebuild that and then we'll go back to the application in the browser and see that it all works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=384.5) [So, back in the browser, let's refresh our legitimate request and make sure that that still works out okay](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=391.5) [and it does and now we will actually change that for our malicious request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=397.5) [and we get the exception. So, this is great and now what we're getting is we're getting this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=404.5)[System.ApplicationException, which we just manually raised. It hasn't even gone all the way through to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=408.5) [the database so we've actually kept this exception in the web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=414.5) [Again, we'd handle it more gracefully. We would say that there is something wrong with the request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=418.5) [We probably wouldn't throw an internal exception, but this is a good example of how a whitelist has been able](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=422.5) [to handle this out-of-bounds sort of condition rather than actually letting that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=428.5) [malicious payload go through to the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=9&mode=live&start=434)

[Demo: Entity Framework’s SQL parameterisation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live)

[So, we've looked at parameterizing our inline SQL and we've looked at using stored procedures.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=0) [Another really good way of mitigating the risk of SQL injection is to use an ORM or an Object Relational Mapper.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=6) [Now in the ASP.NET world, the ORM that we probably most commonly use would be Entity Framework.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=14) [So, let's go through and implement Entity Framework in place of the current stored procedure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=20) [So, what I'm going to do is jump into my Solution Explorer and add a new item and we will find this item](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=26) [under data and it is an ADO.NET Entity Data Model and let's make this InjectionModel.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=34)[Now we'll go to the next page, we will save this data in the connection string for now, and we will call it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=45) [InjectionEntities. And what we will do is just select the entities that we need.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=51) [In this case, I'm only going to do the Product table and Finish. (Typing)](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=61) [Okay, so now I have my entity and I can go and bind that up and actually query directly against this data context](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=72) [rather than going to a stored procedure. So, now let's go and remove everything that isn't around binding up](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=79) [that grid view and we'll get rid of all the manual construction of the connection and calling the stored procedure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=88) [So let's just create a new data context and these were InjectionEntities and what we should do down here is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=97) [let's just bind dc.Products and we can write a nice little where clause here with a lambda statement,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=108) [p.ProductSubCategoryID will be equal to the id from earlier on and we'll turn that into a list and rebuild.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=117)[Okay, so let's go and see how that works in the browser. Alright, so let's start out by refreshing the count page](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=136) [and it works fine. Now, of course, all of this is happening after our whitelisting so even if we do try and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=145) [attempt any injection, it is not going to work. So, that all works fine, but let's go and have a look at why](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=151) [Entity Framework offers us some SQL injection protection. So, what I'm going to do is just resize this window](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=157) [and I have SQL Server Profiler running just behind there. Let's now refresh this page and see the query that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=163) [has actually been generated and executed by Entity Framework and we can see one query just here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=171) [So let's take a bit of a close look at that and then what we'll see is the same sort of parameterization as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=177) [we implemented earlier with the inline SQL. So, this is the beauty of ORMs; they implement that parameterization](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=183) [for you automatically. There are obviously many other reasons why you might also use an ORM, particularly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=191) [around productivity. From a SQL injection perspective, they're particularly good because they do automate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=197) [that parameterization and offer fantastic protection against malicious requests.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=10&mode=live&start=203)

[Demo: Injection through stored procedures](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live)

[One thing about stored procedures that does often tend to be a little bit misunderstood is that there is an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=0) [assumption that stored procedures aren't vulnerable to SQL injection at all and that's not quite the case.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=6) [It is still possible to get SQL injection through to a stored procedure, even when the ASP.NET parameterizes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=12) [the untrusted data. So let's have a look at how that might work.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=20) [I'll go to Search page up here and I'm going to search for term, say, crank and we have 6 results.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=25) [Now what I'm going to do is add a little bit of SQL injection to here, I'm going to add a single quote and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=36) [shortly we'll see why we're doing that and then I'm going to add our classic or 1=1 condition and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=41) [I'm going to add a couple of dashes, which is normally TSQL syntax for commenting and, again, we'll see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=48) [why we do that in a moment and let's now run that query and we've got 504 products.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=54) [This is showing us that we have been able to execute SQL injection and it's actually going through a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=62) [stored procedure so let's have a look at what's wrong with the procedure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=68) [So back over in SQL Management Studio and we have a SearchProducts stored procedure so let's have a look at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=73) [what is in this stored procedure. Now what we'll find here is that it is taking in a SearchTerm and it is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=79) [actually declaring the query as a local variable and it is concatenating SQL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=89) [So, in fact, what we're really doing is we're moving that risk of concatenating query with untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=95) [from the web application, which is where our first demo was, and we've moved that risk back into the stored procedure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=101) [Now all of this gets concatenated and then it gets called with an EXEC statement.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=108) [Now what that means is, is that we can still get our malicious payload through this parameter.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=114) [So, if you think back to that SearchTerm from before, let's have a look at how that would've worked once it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=121) [hit the stored procedure. So, in fact, if we take the value of this and we'll just speculate a little bit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=126) [about how that would've looked, we can remove the double quotes, which is just for quote escaping, so what](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=132) [ended up happening was we had crank look like this and then it closed off with a single quote, if you remember](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=138) [we added a single quote or 1=1, and then it commented out everything beneath there.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=145) [So, ultimately, we ended up with a query that looks just like this and then obviously that resulted in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=151) [quite a large number of records, so we were still able to get that payload through.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=156) [Now, of course, the way to mitigate this within a stored procedure is preferably not to do any concatenation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=162) [of the query and of the untrusted data, but we can still make this query secure with inline SQL, we just need](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=168) [to apply the same mitigations as we did back in the web application, which was to make sure that there's a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=176) [separation of the query and the untrusted data and we'll do that for our parameterization using the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=181) [sp\_executesql function. So let's get rid of the statement that we just ran before because we won't need that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=188)[and let's change this to an ALTER and now we're going to try and improve the security of this query.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=195) [Now the first thing we'll do is that when we run sp\_executesql, the query that we build up needs to be an NVARCHAR.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=204) [Secondly, what we're going to do is when we build up our SQL statement, rather than concatenating the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=213) [SearchTerm, we're going to refer to the SearchTerm as a variable within the statement.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=219) [So, we're just going to encapsulate our percentage symbols in quotes and we'll do a double quote,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=224) [which will escape to a single quote, so that's just fine, and now we will go down and instead of just executing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=230) [the query, we will execute sp\_executesql. Now, when we call that, the first value that we pass through to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=237) [that function is going to be the SQL query. The second value is going to be a list of parameters that we pass](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=247) [through to that query and in this case it's just going to be 1 and we have to pass an NVARCHAR so we'll start](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=255) [that with N and we'll enter SearchTerm and we need to declare a type, which is of course a varchar(50).](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=260) [Now the final value that we're going to pass through to that function is actually an assignment of a value](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=271) [to the parameter and, of course, the parameter name is SearchTerm and the value that we're going to pass through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=276) [to the parameter is the one that comes into our stored procedure, which is also named SearchTerm and that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=284) [should do it. So let's now run that so we update the procedure and we'll go back to the web application and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=293) [see if it works. Okay, so here's our old malicious query, which resulted in SQL injection and every product](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=300) [in the database being returned. Let's now take out the SQL injection and just see if it works normally.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=307) [Okay, great, it does - we get 6 products. Now let's go back and apply that same SQL injection statement again](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=313) [and see what happens and we get 0 products. Now the reason why we get 0 products is now it is trying to match](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=320) [this entire term so rather than a single quote breaking out of the data context and entering the query context,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=328) [it is now assessed as part of the search term. So, the query is actually looking for product names that are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=336) [like crank' or 1=1-- and, of course, none exist so we get 0 results and that is exactly what we would expect](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=342)[to see from an application that does have a robust mitigation against SQL injection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=11&mode=live&start=351)

[Demo: Injection automation with Havij](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live)

[So, the last thing I wanted to talk about in this module is to take a bit of a look at just how easily](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=0) [web applications can be exploited using automated tools. Now this is important because automation lowers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=5) [the bar to entry and what you'll see in a moment is just how easy it can be for someone with no security](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=13) [knowledge whatsoever to actually maliciously attack a site via SQL injection and the example I want to use today](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=19) [is a tool called Havij and Havij is a product from a legitimate company called ItSecTeam and they positioned](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=27) [this product as something that can help penetration testers to find and exploit SQL injection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=35) [Now, of course, the reality of it is it can also help people who are not penetration testers find and exploit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=42) [SQL injection vulnerabilities and they even make a version of this product available for free.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=49) [So, you can imagine that it is very attractive, particularly to this genre that we now know as hacktivists](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=55) [because it is so easy for anybody to get on board and start exploiting websites.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=60) [So let's take a look at what Havij is and how we can use it against](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=66) [the application that we've been using throughout this module.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=69) [Now before preparing this exercise, I reverted all of the good work we did in protecting the application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=72) [so we now have another vulnerable application to test Havij with.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=78) [So let me drag Havij over and we're going to take that URL that we just had and we're going to paste it in place](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=82) [and the first thing that Havij lets you do, and obviously all of this is done by our nice little graphical](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=91) [user interface, is it lets you analyze a target URL. Now when we hit analyze, it's gone away and discovered](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=96) [some pieces of information about both the web application and about the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=104) [Now what we're going to do is ask Havij to have a look at the tables in the database and we'll see it already](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=111) [knows the database name, it knows it's called 1-Injection. We're now going to ask it to get the tables for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=116) [that database and you can see, it's just gone off and it's found this information extremely quickly.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=123) [It is fast because it's running locally on my machine, but also Havij is extremely efficient at actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=129) [getting data out of a database. So let's say we want our data out of the CreditCard table.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=135) [We'll just check CreditCard and we'll then ask it to get the columns and here are our columns.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=141) [You may recognize some of these queries down here because it's a similar sort of approach to what we used](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=149) [earlier to manually execute SQL injection on this side. Okay, so let's now go and ask for the data and I](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=154) [would like to get the CardNumber, the CardType, the ExpMonth, and the ExpYear, and then it's just simply](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=161) [a matter of clicking on Get Data and as we can see, it's now going off and retrieving obviously quite a large](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=169) [number of records. Now the reason why this tool is so effective is because it is so easy to use and there are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=175) [a lot of tutorials on the internet, particularly on YouTube, of people demonstrating how to use Havij against](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=184) [live websites. Many of the times, particularly of the YouTube videos, it's very easy to tell that when we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=192) [say script kiddies, we often are literally talking about kids.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=199) [It is such an easy tool to use that you don't need to have any security knowledge whatsoever to get in and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=203) [exploit a vulnerable website. A lot of the tutorials also include information, such as Google Docs searches,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=211) [so searches that allow you to do things like say, give me a list of maybe ASP.NET websites that have a URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=219) [containing a parameter called id= and obviously that's the sort of thing which is commonly vulnerable to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=228) [SQL injection. So, Havij makes that extremely easy and it gives you a really good idea of just how risky](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=234) [and accessible SQL injection is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=12&mode=live&start=242)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live)

[So, in Summary, this risk is a really good example of security in depth, where we apply mitigations at multiple](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=0) [different layers of the application. So, beginning with the database, thinking about what are the bare minimum](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=6) [set of permissions that the web application actually needs in order to do its job.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=13) [Could it possibly have multiple different accounts for the different roles in the application to even further](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=18) [reduce the risk? Always parameterizing untrusted data, so never just concatenating the query and the data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=24) [provided by the user. It's really important that if we're writing inline SQL that it is parameterized.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=32) [Of course, stored procedures are probably a better mechanism than inline SQL for many reasons anyway,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=39) [plus they provide us the native ability to parameterize, but as we saw, we've got to remember that we can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=45) [still pass an injection payload through to a stored procedure and the internals of the stored procedure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=50) [can still succumb to the same risk. Always validating our untrusted data against a whitelist and this is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=56) [going to be a recurring theme in subsequent modules. Making sure that we use things like type conversion](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=62) [if we can perhaps filter out anything that isn't an integer that should be.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=69) [Applying regular expressions, say, to emails or phone numbers or anything that complies to a fairly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=74) [predictable set of characters and also, looking at maybe using known good values.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=79) [In short, making sure we have some way of providing some form of security validation around the data that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=86) [enters the system before we start processing it. Using ORMs because they have a native ability to parameterize.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=92) [We saw how Entity Framework applied all the parameterization logic that we had manually done with inline SQL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=100) [So, they're a great timesaver, they're a great automation feature for doing data excess, they're also a great](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=107) [security feature. And finally, remembering the tools like Havij are freely available and are very, very good](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=112) [at very quickly exploiting a vulnerable application by somebody with little or no security knowledge whatsoever.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=120) [It makes hacking vulnerable applications really accessible to just about anyone and, as we saw,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=128) [the consequences can be really severe.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m1-injection&clip=13&mode=live&start=134)

[Cross Site Scripting (XSS)](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live)

[This is Troy Hunt and welcome to Part 2 of the OWASP Top 10 on Cross Site Scripting or as we kindly refer to it, XSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=0) [In this module, we're going to start out by having a look at how OWASP views the risk and then we'll move on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=11) [to actually performing an attack and exploiting an application at risk of XSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=17) [We'll then delve a little bit further into XSS and understand how the risk manifests itself in a web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=24) [We'll move on to an overview and an implementation of output encoding in both HTML and JavaScript.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=31) [Then we'll also take a look at the native output encoding defenses in both Web Forms and MVC.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=37) [Once again, we'll look at whitelisting as a means of keeping untrusted data sanitized in the system and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=44) [we'll also talk about using the native request validation in ASP.NET as a defense against malicious input.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=50) [We'll take a look at reflective versus persistent XSS and how they differ in execution and we'll take a look](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=58) [at the native defenses offered by the browsers because they do differ quite a bit.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=64) [And finally, we'll talk about some of the approaches an attacker uses to get XSS through to a victim,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=69) [including payload obfuscation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=0&mode=live&start=75)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live)

[Taking a look at the OWASP overview and risk rating, once again the Threat Agents can be pretty much anyone](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=0) [who can send untrusted data to the system. So that's obviously external users, but it could be internal users](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=6) [and administrators as well. Once again and as with SQL injection, this attack is a text-based attack.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=13) [In the case of XSS, it is exploiting the interpreter in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=21) [The attack can come from various sources, obviously from external parties, but it can also come from internal data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=26) [sources within the application and we'll talk about that a little bit when we look at persistent XSS later on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=33) [XSS is extremely prevalent and OWASP actually views it as the most prevalent security flaw.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=40) [The detectability though is very easy. It is very simple to find applications that have cross site scripting](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=47) [vulnerabilities. The Technical Impact is moderate. It's not as severe as something like SQL injection where](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=54) [an attacker could gain immediate access to internal data, but certainly XSS can be used for very malicious](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=60) [purposes such as hijacking the sessions of unsuspecting victims.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=66) [Once again, the business impact is always going to be dependent on the nature of the attack and just how much](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=71)[damage an attacker is able to do with a cross site scripting vulnerability.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=1&mode=live&start=76)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live)

[Let's now take a look at what's involved in exploiting application at risk of cross site scripting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=0) [What we're going to do is use a sample application designed specifically for the purpose and this sample application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=6.5) [has a Search feature on the front page so this is obviously a pretty typical feature within a web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=13.5) [Now what we're going to do is search for lager.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=19.5) [When we search for lager, we have a number of results, but what I'm really interested in here is two things.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=25.5) [#1 is that we can see our search term here in the query string and #2 is that we can see our search term repeated](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=31.5) [to the page, it's reflected to the page if you like. Now to start investigating whether we might have a cross](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=39.5) [site scripting risk, I want to start manipulating this query string and what I'm going to do is I'm going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=45.5) [put some HTML in the query string and, in fact, what I'm going to do is wrap the word Lager in italics tags.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=52.5) [(Typing) Now let's see what happens when we load that page. Okay, well there's two things.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=59.5) [So, going from the bottom, there are no results found, which is not surprising because we probably have no lagers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=67.5) [in the database that are wrapped in italics tags. What's a bit more interesting, though, is that when our search](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=73.5) [term is repeated to the page, the search term now appears in italics.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=80.5) [So rather than the website telling me that I've searched for lager encapsulated in an italics tag, it has actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=86.5) [put the Lager search term in italics. So, what we've done is we've actually changed the markup of the page and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=94.5) [this is one of the first things an attacker will do when they're looking for an XSS vulnerability.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=101.5) [Now to try and understand the risk a little bit better, let's take a look at the source code of the page and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=107.5) [see what we can see. So, this is a fairly typical Web Forms page and if we scroll down a little bit, we will see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=111.5) [in a header tag our search term and, in fact, the search term itself is right here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=122.5) [Now when we look at the HTML here, this looks like perfectly valid markup, the italics a perfectly valid markup.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=129.5) [The difference is, is that everything that is highlighted just here is actually data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=137.5) [Now what's happened is the data that we've entered has managed to manipulate the markup so we've actually broken](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=142.5) [out of the data context and we've also entered the markup context and that's why these italics tags have been](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=148.5) [rendered and that's why the text appears in italics on the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=154.5)[So one of the big things with XSS is we need to make sure that this cannot happen.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=159.5) [Let's take a look at what sort of damage can be done by manipulating the markup via XSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=165.5) [Okay, so I'm going to close the Page Source and I'm going to go and manipulate the query string and we're going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=172.5)[try something a little bit more malicious, but before we do that, I want to show you one thing and what I want](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=179.5) [to show you is some cookies that are in the browser. So, what we're going to do is jump down to the Developer Toolbar.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=185.5) [Now you'll also notice that we're doing all of this in Firefox and you'll see why a little bit later on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=193.5) [Now what we're going to do is use Firefox's Developer Toolbar and, in fact, the Developer Command Line here](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=198.5) [to get a list of all the cookies we have. So, we'll just do a cookie list and we will see that we've got one](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=204.5) [cookie so the name of this cookie, the key, is SecretCookie. The value of it is "Shhh……it's a secret".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=211.5) [So, we have a cookie which exists on this website, right. So keeping in mind that we have this cookie set for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=220.5) [this website and the domain of this website is XSS, it's a host entry on my local machine, let's change the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=227.5) [query string with something a little bit more malicious. Now let's have a look at what this query string says.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=233.5) [We've still got the search term Lager, but now we've got a script tag so we're now going to try and execute some](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=240.5) [JavaScript via XSS. The JavaScript is going to change the location of the browser and it's going to direct it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=246.5) [over to evilsite, this is a different domain. It's going to load a path on evilsite called CookieStealer.html](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=253.5) [and the really important bit here is it's going to pass a QueryString which appends document.cookie,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=263.5) [which is going to be any cookies that the browser has access to on the current domain, on the legitimate domain.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=269.5)[The %2B is an encoded plus symbol. So, let's load this and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=275.5) [Okay, now this is really important. So, we can see a couple of things here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=283.5)[Obviously we have a JavaScript alert box in the middle of the page and this alert box is mocking us a little bit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=286.5) [and showing us that it actually has the cookie. We can see that SecretCookie is there and we can see the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=292.5) [value of SecretCookie. The important thing is that this is evilsite, so this is a totally different domain.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=298.5) [Now cookies are not meant to be accessible across domains in this way and it's only because we're able to exploit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=305.5) [the cross site scripting vulnerability that the evilsite was able to access the cookies from the legitimate site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=312.5) [Now what that means is this -- if we go back and we look at that path that we entered just before,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=320.5) [if we could find a website that was vulnerable to cross site scripting in the way that this one was and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=328.5) [we could have someone who had a cookie on that website follow a link that was very carefully constructed like this,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=334.5) [we could steal their cookies and we could do this because this website has let us.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=340.5) [This website has let us run arbitrary JavaScript in the browser, which was able to access cookies and then send it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=345.5) [to somewhere else and that's the real power of XSS, the ability to run script, access objects in the DOM, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=352.5) [send them over to somewhere else.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=2&mode=live&start=360.5)

[Risk in practice: My Space and Samy](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live)

[As with all the risks in the OWASP Top 10, it's always good to try and tie them back to an industry precedent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=0) [so that we can see that they're really not just theoretical, they do actually get exploited in real life.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=6) [A very good example of XSS being exploited and arguably the first time it was exploited on a very large scale](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=13) [with serious impact was Sammy's attack on Myspace in 2005. Now what Sammy was able to do was write a script and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=19) [here's a transcript here that explains the whole process, but Sammy wrote a script and it was a self-propagating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=29) [cross site scripting worm. Now what this meant is that on Myspace in the space of only 5 hours, he was able to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=35) [cause over 1 million people to automatically befriend him and this was simply because he was able to get the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=43) [script to self-propagate. So, it was an XSS worm. Now what's interesting about this XSS worm as well is just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=50) [how complex it was. So let's take a look at the code behind his worm.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=59) [Now I'm not going to look at this detail, but it's enough to give you an idea of just how complex cross site](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=64) [scripting can get. So here's his code. We just looked at examples of turning text into italics or stealing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=70) [someone's cookies. Have a look at how advanced a cross site scripting attack can get.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=76) [There are numerous techniques in here to bypass cross site scripting filters, to bypass defenses built into the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=83) [application, by employing a whole bunch of little tricks that ultimately we were able to get him that worm that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=89) [could self-propagate so quickly. So this is a really good example of just how serious cross site scripting can be.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=3&mode=live&start=96)

[Understanding XSS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live)

[Let's now take a closer look at what makes cross site scripting possible and where I'd like to start is by having](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=0) [a look at the URL. So let's take a typical URL and this is a similar sort of URL to what we just saw in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=6) [example we exploited and, once again, in this URL obviously we have a domain, we have a path, which is Search,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=13) [and we have a query string, which is Lager. Now, let's consider how that then translates through into the markup.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=20) [So in this case and in the demonstration we just did, the word Lager is repeated onto the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=27) [Now if you think back to when we looked at SQL injection, one of the concepts we identified was a Trusted resource.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=34) [Now in the case of cross site scripting, we trust the domain, we trust the path, we do control the web application,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=41) [and then in terms of the output, we trust the welcome text, so you searched for and we trust the strong tag.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=48) [We have hardcoded this into the markup. Obviously we've also hardcoded the closing strong tag.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=56) [Now what we don't trust is the user input. So, once again, this is Untrusted data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=62)[So in this case, the word Lager, which in reality is just the value of the q query string, this is Untrusted.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=69) [So, again, we've got to go back and think that we cannot verify the integrity and we must expect that the intent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=76) [could be malicious. Now what that means is, is that we have to think about how to mitigate the potential risks](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=84) [that can be introduced by writing that text to the page. Let's look at a few different ways in which we can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=91) [mitigate that explicitly by coding and some ways that we can mitigate it implicitly using ASP.NET's native defenses.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=4&mode=live&start=97)

[Output encoding concepts](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live)

[The single most important thing to understand with cross site scripting mitigations is the concept of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=1) [outputting coding. When we searched, we expected that search term Lager to only be data, not markup.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=7) [We never intended it to actually change the behavior of the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=16) [Now it did that first of all by changing the rendering of the text and second of all by running script in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=21) [Now all that was only possible because, as I mentioned when we looked at the source code, the untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=27) [was able to break out of the data context and into the markup context so it was actually able to change the way](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=33) [the page was structured, not just from the data, but the way the markup was built.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=40) [Now the way to mitigate this is that we want to make sure that that untrusted data is rendered back to the screen](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=45) [exactly the way it was entered, not rendered to the source code exactly the way it was entered.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=53) [So, what we need to think about is how do we get the word Lager wrapped in italics tags to actually still appear](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=58) [on the screen wrapped in italics and the answer is simply HTML encoding and this is probably pretty familiar](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=65) [to most people. So, in HTML, if we want to display a less-than symbol in the browser, we use the escape character,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=73) [which is ampersand lt semicolon. If we want a greater-than symbol, it's ampersand gt semicolon.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=81)[This is what output encoding is. It's ensuring that we can take those input characters and still render them](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=89) [to the screen by encoding them correctly. Now if we can do that and it appears in the HTML source escaped,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=96) [then it can't enter the markup context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=103) [Now one of the things that we have to remember about output encoding is that we have many, many different contexts](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=107) [which we might need to encode for. So, we were just looking at the HTML context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=113) [So in the HTML context, a less-than symbol is ampersand lt semicolon, that's the escape sequence.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=118) [It is a different escape sequence in JavaScript, it's a different escape sequence in CSS, and in fact,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=126) [there are many different possible output context where the encoding will differ.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=133) [How we encode for XML, for example, is going to be different and it's very, very important to remember when we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=138) [are doing output encoding that we output encode for the correct context because if we don't get that right,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=144) [we will have remaining vulnerabilities. So let's look at an example of how that encoding changes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=152) [So, again, going back to our original string of the word Lager wrapped in italics tags, you can see here that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=157) [for the three different contexts, which you almost commonly deal with, HTML, JavaScript, and CSS, you can see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=165) [the escape sequences are very different. Now if we tried to apply the CSS escape sequence to untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=172) [that was embedded in an HTML page, it would not render correctly. We would get an entirely different result.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=180) [We would also have a remaining risk because it wouldn't actually mitigate the potential for cross site scripting](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=187) [to be exploited, it's not the correct escape sequence. So, again, and we'll look at this in detail in just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=192) [a moment, it's extremely important to get the right output encoding context depending on where we're actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=198) [embedding that untrusted data, but one thing that's extremely important when you are doing output encoding is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=205) [to use a proven library. You don't want to be implementing your own encoding process that individually looks for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=212) [characters and converts them to the correct output sequence depending on the context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=220) [Not only would that be a very laborious process, you can introduce security risks by trying to implement a process](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=226) [that can be quite complex. Use proven library and we'll look at some of those available both in the native](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=233) [ASP.NET framework and available via NuGet in some of the subsequent exercises.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=5&mode=live&start=240)

[Demo: Implementing output encoding](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live)

[Let's start to look at how we can protect this vulnerable application from the risk of cross site scripting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=0) [Now we're looking at the code behind of the Search Page and we can see a couple of things happening here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=7) [So, first of all, we can see our q QueryString being red and assigned to the variable of searchTerm and then we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=12) [can searchTerm being set as the text attribute of a searchTerm control and we can see here that control is a label.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=19) [Now what we need to do is make sure that when that searchTerm text is set, we are setting it to an encoded value](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=29) [of the untrusted data and to do that, we're going to use a method out of the AntiXSS framework.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=36) [Now as of .NET 4.5, we have the AntiXSS framework built into the .NET framework.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=43) [Now we can add a reference to the AntiXSS Libraries by using System.web.Security.AntiXss.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=50) [Now that's going to give us access to a very important method within the framework and that method is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=61) [AntiXssEncoder.HtmlEncode and we're going to wrap this method around our searchTerm before we assign it to the label.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=69) [Now this particular method requires two parameters and the second parameter is a Boolean to indicate whether](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=78) [or not we use named entities. Now named entities will do things like take a copyright symbol and convert it to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=85) [ampersand copy semicolon. So, it's a very intuitive way of actually output encoding our entities so we'll turn](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=93) [that on to true. Okay, so let's now rebuild that application and we'll see what happens when we try and run it again.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=100) [So, here we are back in Firefox and the page is still loaded where we had it before, where we were actually able](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=107) [to manipulate the markup context and let's refresh it and see what happens now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=113) [Okay, now this is a very different result because what we're now seeing is we're now seeing that word lager](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=120) [wrapped in italics tags. We're not seeing it rendering italics; we're actually seeing the italics tags rendered](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=126) [to the page. So let's have a look at what has changed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=132) [We'll take a scroll down of the source code and what we'll now see is that that italics tag has been escaped and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=139) [this is exactly what we just looked at in the slides where we saw a less-than symbol escaping to ampersand](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=146) [less than semicolon. Now this is very important because it means we cannot enter that markup context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=153) [It's just going to render to the page exactly as we entered it. We will not be able to do anything malicious](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=160) [with cross site scripting the way this has been output encoded.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=166) [So that's great, but now let's go and take a look at what happens if we want to output encode](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=170) [the untrusted data for the JavaScript context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=175) [So, we're back over to the project and I've opened up the Search.aspx Page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=180)[Now what I'd like to do is actually implement some output encoding for the JavaScript context and, in fact,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=185) [what I want is I want to have a script tag here and of course it'll be a type of JavaScript and, in here,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=191) [I'd like to assign a variable and we can call this q and I would like to assign this to the value that is passed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=198) [in through the QueryString, but I want to do it securely so I'm going to need to implement some output encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=207) [Now before I can do that, we do need to go out to NuGet and get the external AntiXSS Library, the library that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=214) [is external from the framework. The one in the framework doesn't implement the JavaScript output encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=221) [So let's jump over to Solution Explorer and we will go to our NuGet Packages and we will go online and look for AntiXSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=227) [Okay, so this is the framework that we want so let's just install that guy, accept the license.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=241) [Now this is also important for anyone who's working with pre-ASP.NET 4.5 and wants to use the HTML encoding](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=249) [features of the AntiXSS Library. Okay, so now we can actually start to code against that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=256) [So, we are going to call into Microsoft.Security.Application.Encoder.JavaScriptEncode and we are going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=263) [request from the QueryString, the value that is equal to q and we just give it a quick rebuild and back in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=282) [browser let's just give it a refresh. Everything aligns with that arrow.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=296) [Let's now take a look at the Page Source and see how that string has been encoded for the JavaScript context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=300) [So, if we scroll down a little bit and we can see our JavaScript context and here's our assignment of the q variable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=307) [and, in fact, what we can see is that we've got a very different encoding so this is now our JavaScript encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=316) [There's our less than symbol, there's our greater than symbol, the lager search text right in the middle,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=322) [and then the whole thing in reverse and, of course, we have got a front slash here in order to close off the tag](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=329) [and another greater than symbol, but the important thing is, is that the output encoding is very different to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=338) [the output encoding of the HTML. We have encoded correctly for the correct context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=6&mode=live&start=344)

[Demo: Output encoding in web forms](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live)

[In the example we just looked at, the searchTerm, the untrusted data was output to the text attribute of an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=0) [ASP.NET Web Forms label control. Now, that did not output encode automatically; we had to manually do the output](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=7) [encoding and one thing that's very important to understand is that there are many different ASP.NET Web Forms](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=15) [controls with many different attributes and the output encoding is very, very different between them.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=23) [Now what I mean by that is there are many controls that do implement output encoding natively and there are many](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=30) [that don't, such as the text attribute of the label control, and that's the other thing to remember.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=36) [It's not just the control, but it's the individual attributes and we'll have a look at that in a moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=41) [This is a list of Web Forms controls and their properties and then whether h1 will automatically HTML output encode](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=47) [or not. There are over 1,000 entries just here and what you'll see in that file right-hand column is that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=57) [there's a lot of trues and there's a lot of falses so a lot of these will not automatically output encode.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=64) [There are 34 pages here so it is a very, very broad landscape of native output encoding versus no-native](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=72) [output encoding. So let's take a look at it in the browser and get a bit of an idea about how different the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=80) [implementation can be with different controls and different properties.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=86) [Back on our familiar application, let's have a look at some of the idiosyncrasies of Web Forms encoding and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=91) [I've got four examples here. Now the first example is a GridView with a BoundField.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=99) [Now what I've done here is bound a collection, which has our now familiar XSS payload, which is Lager wrapped](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=104) [in italics tags, and what we'll see here is that a BoundField will correctly output encode.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=110) [So, we've got the italics tags wrapped around the searchTerm and we don't actually have this text breaking out](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=117) [into the markup context. However, a GridView with a HyperLinkField gives a different result.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=124) [So, HyperLinkField will not automatically output encode and that's why we can see that it is in italics now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=131) [It's also underlined because a HyperLinkField is, of course, a link.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=139) [A Button control with a Text attribute, so the Text attribute of a Button control is clearly output encoding](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=144) [because, again, we've got the markup wrapped around the Text rather than rendered to the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=151) [Conversely, and we already knew this, a Label control will not automatically output encode and that's why Lager](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=157) [appears in italics. What's interesting with this one is that if we hover over here, we can see there's a tool tip.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=163) [The tool tip does automatically output encode. So, even within the one control, some properties will implement](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=170) [automatic implicit output encoding and others will know. What this all means is that when you are working with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=177) [Web Forms, you've got to be extremely cautious about testing what is being output encoded and what is not.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=184) [You cannot just assume that output encoding is happening for everything because for many, many controls and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=193) [many different properties, it's not and you have to go through and implement it manually.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=199) [In many ways, that actually makes Web Forms somewhat less secure than ASP.NET MVC.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=205) [So let's go and see how MVC handles implicit output encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=7&mode=live&start=211)

[Demo: Output encoding in MVC](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live)

[The really neat thing about ASP.NET MVC is that the Razor View Engine automatically HTML encodes all output](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=0) [by default. You actually have to explicitly tell it not to in order to have a cross site scripting vulnerability.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=8) [So let's take a look at how that looks and what I'm going to do is let's just take a p tag and we'll take an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=16) [at symbol to start our Razor syntax and what we'll do is we will just put a string value, which is going to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=22) [our now very familiar XSS test pattern and we'll close that off and save that view.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=30) [Now this is all in the same ASP.NET Web Forms application, it's just that we've got an MVC controller and view](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=40) [in order to test the MVC behavior. So let's jump over to the browser and see what this looks like.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=46) [Okay, back in the browser, let's go and have a look at the MVC encoding test.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=52) [Okay, now what we can see is exactly what we would expect to see with properly encoded markup.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=57)[We can see the italics tags rendered to the screen. Now this is just pure Razor syntax.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=63) [Let's go and have a look at what happens with an HTML helper.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=68)[Okay, so I'm just going to duplicate this line and we will do an html.Label.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=72) [Let's save that page and we'll flip back and see what happens with an HTML helper.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=80) [Okay, so back on the test page and we Refresh and we've got exactly the same result because the HTML helper](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=85) [also automatically output encodes. It's a very implicit behavior, but what if you really wanted to avoid](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=91) [output encoding? What if you didn't want it to happen at all?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=98) [So, for example, what if you had a database that might be part of a content management system and you wanted](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=101) [to store markup in the database and you wanted that markup to actually render to the source code and not just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=107) [render to the browser? So, in effect, what would happen if I wanted to store Lager in the database and display it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=113) [to the screen in italics? Now this is where the raw HTML helper comes in. So let's go back and see how that works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=120)[So, I'm just going to duplicate that last HTML helper and we're going to change Label to Raw, Save that, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=128) [let's go back to the browser and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=135) [Alright a quick Refresh and this time we see Lager in italics.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=140) [Now this is really important because what it's showing is that ASP.NET MVC is giving us a secure by default position.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=144) [Now what I mean by this is that the Razor syntax and the HTML helpers are automatically HTML encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=153) [The only way we could actually get data to enter the markup context is we had to very explicitly tell it to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=160) [by using the Raw HTML helper. Now in many ways that actually makes things more secure than ASP.NET Web Forms,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=166) [where it's very easy to make a mistake and not realize that a particular control or property of a control is not](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=174) [automatically output encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=8&mode=live&start=181)

[Demo: Whitelisting allowable values](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live)

[So far all the mitigations that we've implemented to handle cross site scripting](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=0) [risks have been about accepting the input and then trying to handle it in a graceful fashion.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=4.5) [One thing we touched on back in the SQL injection exercise was about validating all untrusted input against a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=10.5) [whitelist of allowable values and we looked at different ways of implementing whitelists and one of the most](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=18.5) [effective ways of implementing a whitelist against a string input is by using a regular expression.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=24.5) [Now in this case, when we're performing a search, there's a fairly finite list of expected characters that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=31.5) [would want to allow for a search. Now as we discussed back in the module about SQL injection, you do have to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=37.5) [very careful that you don't exclude any legitimate untrusted data by an overenthusiastic regex.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=43.5) [What we're going to do is implement a very basic one here and obviously it's something that you'd probably](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=52.5) [tailor a little bit more depending on your use case.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=56.5) [So, we're going to start out by using System.Text.RegularExpressions and the point that I want to validate our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=59.5) [untrusted data is immediately after we receive it. So right about here, now what I'd like to do is make sure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=70.5) [that that searchTerm matches our Regex. So, we're going to wrap an if clause around here and if we don't match](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=78.5) [our particular Regex and it is going to be a match and we'll apply that match against the searchTerm and the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=84.5) [string that I'm going to use for my Regex with a pattern looks like this.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=92.5) [Now this is a fairly basic one and what's it's going to do is look for language characters, as well as looking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=97.5) [for spaces, dots, and dashes. You probably want to be a little bit more liberal and include things like quotes,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=104.5) [which is a very common character that's excluded when it usually shouldn't be, particularly for things like names,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=110.5) [but for a basic example for Search, this will do. Now if it doesn't match, we are just going to throw a new](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=115.5) [ApplicationException and, again, it's always better not to throw an exception, but to handle it a little bit more](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=122.5) [gracefully, but this will give you an idea of how the whitelist works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=128.5)[In this case, we would just say "Search term is not allowed."](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=132.5) [Okay, let's build that and let's see what that does to our Search function now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=137.5) [Okay, back on the Search page, let's search for lager again and I'll search for lager, a legitimate lager term,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=142.5) [does exactly what it did before. Now let's wrap our italics tag around that search term and see how our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=151.5) [whitelist affects it and that's great. We have had an exception thrown.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=158.5) [Now, again, we wouldn't normally throw an exception, we'd handle it more gracefully and we wouldn't return](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=163.5) [a yellow screen of death like we see here, but it does show that we've been able to catch that untrusted data,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=169.5) [which hasn't passed our whitelist validation, very early on. So that concept of validating untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=174.5)[against a whitelist of allowable values is extremely important and that's why you keep seeing it come up](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=182.5) [across the different modules.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=9&mode=live&start=188.5)

[Demo: ASP.NET request validation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live)

[In order to demonstrate the risk of cross site scripting in this module,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=0) [I actually had to change one of the secure defaults that ASP.NET comes with and that secure default is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=5) [requestValidation. Now what requestValidation is, is it is a native implicit defense that looks at all](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=12) [untrusted data that comes into the system, so via means such as form posts and query strings, and it validates](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=20) [that untrusted data against its own whitelist of allowable values.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=28) [Now what that means is that natively, if you attempt many XSS attacks, ASP.NET is going to catch that attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=33) [before it even gets to the code behind and starts processing your own business logic.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=41) [So let's go and turn requestValidation back on and see how it changes the behavior of this page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=46) [Now the requestValidation settings are configured in the Web.config and they are configured on the pages element](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=53) [and you can see here that requestValidation had actually been set to false.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=59) [Now we can either set that to true or we can remove the node altogether, which will return it to the default,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=64) [which is to turn requestValidation back on. So, let's save that and now have a look at how that changes the way](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=70) [our application responds to malicious requests.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=77) [So, we're back on the page where there has been an attempt to exploit an XSS risk by passing in italics tags.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=82) [Let's Refresh the page and now see how it behaves. Okay, so you can now see that we've got a very different error.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=88) [So, it hasn't managed to get through to the point where we implemented our whitelist, which was the previous error,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=96) [and in fact what we can see now is a potentially dangerous request was detected from the client and it's looking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=101)[at the value that we passed through in the query string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=108) [ASP.NET has automatically thrown an HttpRequestValidationException so without any coding, it has looked at that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=111) [request value and it said, hey this is probably not safe, let us not process this.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=121) [So, this is a very powerful native framework feature that will automatically provide a degree of protection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=126) [against cross site scripting. Now what we just did is turned it back on across the entire site, which means](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=134) [that any attempt to post any HTML tags to the site or put them in a query string is going to cause](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=141) [requestValidation to fire. Now there are legitimate times where you do want to post HTML tags.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=148) [So, for example, if you have a rich text editor, which actually posts HTML markup after you change content in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=155) [a WYSIWYG view. We don't have to turn it off for the entire site though; we can turn it off on the page level.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=162) [So let's jump back to Visual Studio, turn it back off for this page, and see how it changes the behavior.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=170) [So, I'm back in Visual Studio and I'm on the Search page. Let's now jump up into the page declaration and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=177) [actually change our validateRequest value to false. Now this is just going to turn off requestValidation on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=184) [this page alone. So, if we do have an XSS risk somewhere else on a completely different page, we're still going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=191) [to have the native ASP.NET requestValidation jump in and try to save us from that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=199) [So let's see what this does in the browser now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=205) [So, when we turn requestValidation back on for the entire site, this is what we got.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=209) [Now that we've turned it off just for this page, let's refresh it again and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=214) [Okay, so now we've fallen back to our whitelist validation and we've got the ApplicationException that we threw](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=219) [when we had illegal characters pass through. So, that's great, we've just allowed a potentially malicious request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=225) [to be made to this page without having to turn it off for the entire site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=230) [Now, of course, we've just been manipulating the QueryString mode.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=235) [So let's go back to the Home page and see what happens if we try to search from there and we are going to try and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=238) [pass our italics again and we'll close that tag off. Let's try and search from here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=246) [Now what we're seeing is, again, an HttpRequestValidationException is being raised because what's happening](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=254) [this time is that the control that takes the search term is causing the validation to fail.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=260) [So, we've got a text field called SearchTerm and that's why we're seeing the error message here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=267) [Now as of ASP.NET 4.5, we actually have a very neat feature available, which will allow us to disable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=272) [requestValidation, not for this entire page, but just for this specific control. So let's go and implement that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=279) [Now the first thing we're going to need to do, because this is an ASP.NET 4.5 feature, is we're going to need](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=288) [to change our requestValidation mode from 2.0 up to 4.5. So, that's the first step.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=294) [So, the next thing we're going to need to do is change the requestValidation mode for the textbox control and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=301) [we can do that by going down to the ValidateRequestMode and disabling it on that control.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=307) [So let's save bot those files, go back to the browser, and see how this changes the behavior.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=314) [Okay, so let's try that again with requestValidation disabled just for this textbox control,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=322)[which again is a feature of ASP.NET 4.5.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=327) [Okay, now that has actually worked. What we're seeing here is we've actually ended up over on the Search.aspx page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=332) [So, back on the Default page we have actually been able to submit the form, submit the SearchTerm with the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=340) [italics tag wrapped around it, and that page has now not caused a requestValidation error because we configured](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=347) [that control not to validate the textbox control. We are, however, seeing an HttpRequestValidationException](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=353) [because now that we rendered .NET 4.5 requestValidation mode, we can't just turn it off on an individual page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=361) [In fact, the only way that we can disable requestValidation for either the entire site or an individual page](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=369) [is by ensuring that the requestValidation mode in the Web.config is set to 2.0.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=376) [So whilst we solved the problem of actually entering potentially malicious script into a textbox control,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=382) [we've now moved the problem to where we're reading it from the query string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=389) [So let's have a look at how we can change this in ASP.NET 4.5.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=393)[The solution to dealing with requests that may cause requestValidation in an ASP.NET 4.5 application is far more](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=399) [elegant than what it was before. So, in the previous example, what we did was we re-enabled requestValidation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=407) [for the entire site and then we turned it off just on the Search page, we added this ValidateRequest="false".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=414) [Now, that is still a very broad brush approach because it's turning requestValidation off for the entire page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=422) [ASP.NET 4.5 allows us to get a lot more granular. So, what we're going to do is delete that altogether and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=431) [we're going to jump back over to the code behind. Now in the code behind, we have a much more elegant way of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=438) [dealing with requestValidation and that way is to access Request.Unvalidated and then access the QueryString.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=444) [So, this unvalidated request is not going to cause requestValidation to fire when we access the QueryString value.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=453) [Okay, so let's rebuild that and then see what it does in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=461) [Now back in the browser, this is what we saw when we were just accessing Request.QueryString with angle brackets](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=468) [in the QueryString and with requestValidation turned off at the page level.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=475) [Now, of course, turning it off at the page level no longer works in ASP.NET 4.5.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=481) [Now that we're accessing Request.Unvalidated.QueryString, let's reload the page and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=487) [Fantastic. So now it has fallen back through to our whitelist validation; we're back to where we've manually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=495) [thrown that ApplicationException. Now this is a very powerful feature because what it allows us to do is say](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=501) [there may be some query string parameters that we would like to validate on this page, but there may be other](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=507) [query string parameters that we don't. So, it gives us far more control over the way we implement our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=514) [requestValidation. Now obviously all of that is in Web Forms. Let's take a look at a little feature within](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=521) [ASP.NET MVC, which allows us to achieve a similar effect. So let's just jump back to the root of this website](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=527) [and we have got a little feature here called Create User Account.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=537) [Now this maps back through to an MVC Controller action and let's imagine an example where we are passing in a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=542) [UserName and a Password. So, for example, at registration time or perhaps at login time.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=549) [So, let's enter a fairly typical UserName and I am going to enter a Password that contains a mix of characters,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=554) [including some non-alphanumeric characters and that's quite important for this example.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=564) [So, let us now submit that and what we can see is that we have another HttpRequestValidationException.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=568) [Now clearly that wasn't caused by the UserName; it was actually caused by the Password and it was caused because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=577) [my Password was <Script>. Now there's absolutely no reason why we should reject a password called <Script>](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=583) [unless, of course, we wanted to be very stringent about using perhaps more characters, perhaps using numbers,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=592) [but certainly we shouldn't be rejecting this password because it contains angle brackets.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=597) [There's really not an XSS risk because you were never going to display that password to anyone again,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=603) [certainly not if you implement your application securely. So let's take a look at what's involved in allowing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=609) [a password to contain any character whatsoever without causing requestValidation to fire in ASP.NET MVC.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=616) [So here's my controller and the route that we were just looking at in the browser was the Create Account action.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=626) [Now what Create Account does is, when it does perform a post and it calls the Create Account action, it passes in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=633) [a model with a type of user account. Now you can see user account is clear down here and UserName and Password](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=640) [are just simply strings. Now what we're going to do is decorate Password with AllowHtml.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=650) [Now when we do that, this particular password is going to be allowed to contain whatever character it likes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=660) [without causing requestValidation. So let's just rebuild that and return to the browser and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=666) [Okay, so here's where we had the problem before. Let's go back and I'm going to reenter troy's UserName and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=673) [<Script> as a Password. Let's now submit that and see what happens and that has worked fine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=682) [It has successfully posted and it's redirected me back to the Home page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=688) [So, that AllowHtml attribute has enabled us to post whatever characters we like in the Password field.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=692) [So, that's a very powerful attribute that does a similar sort of thing to accessing the unvalidated request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=700) [object collection, except this is for ASP.NET MVC and it happens just by virtue of decorating that class attribute](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=706) [with the AllowHtml syntax. Now just one last thing on requestValidation - Always code as though it doesn't exist.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=715) [RequestValidation is not perfect. There are certain XSS vectors which can get through so try and think of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=725) [requestValidation as a safety net. It's a last line of defense. If you've missed the other mitigations upstream](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=733) [of it, including your own sanitization and whitelisting of trusted data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=741) [Don't ever rely on requestValidation alone to save you from cross site scripting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=10&mode=live&start=746)

[Demo: Reflective versus persistent XSS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live)

[So far, all of the cross site scripting and text we've looked at are what's referred to as reflected XSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=0) [Now what that means is, is that the webpage is reflecting via its markup the payload, which is entered either](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=6.5) [into the URL or is posted from another page, but there's another type of very common XSS, which is referred to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=13) [as persistent XSS and what this means is that you've actually got an XSS payload somewhere in the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=21) [The risk, of course, with that is that you don't need any sort of reflection at all from the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=30) [So long as you can load that payload up and render it in the page, the risk can manifest itself.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=35) [So let's take an example of that. In my database, I have a piece of persistent XSS and we're going to find that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=42) [when we search for choc. So let's do a search and here you can see an alert and it says, Persistent XSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=49) [Let's now have a look at how that is rendering in the HTML source.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=58) [Now you can see that we've got Chocolade here and then the rendering has stopped.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=61) [If I dismiss this, we can see the next record come through. So, the XSS is probably sitting somewhere just after](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=66) [this record so we're just going to right-click and we're going to inspect the elements and if you have a look](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=73) [down at our HTML source here, we can see we have the hyperlink, we can see Chocolade, and right after that,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=80) [we can see that we have a script tag and this script tag has the alert which we just saw appear in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=87) [Now let's take a look at what's going on in the Search page and how we can protect it from the risk of Persistent XSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=96) [What we've got here is a GridView and this GridView has a hyperlink field.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=104) [Now as we saw earlier on, the hyperlink field is one of those properties of the GridView which doesn't natively](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=110) [implement HTML output encoding, it's not implicit. We're going to have to do this ourselves manually.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=115) [Now there are a few different ways of doing this. One way is to jump in on the RawDataBound event,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=123) [grab ahold of that hyperlink field, get the text value, HTML output encode it, and then we should've mitigated the risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=130) [So let's jump over into SplitView because it's just a little bit easier to bind events from the WYSIWYG view,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=138) [we'll jump over into Properties, into Events, and we'll look at the RawDataBound event, give that a double-click,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=144) [and now we have an event in our code behind. So, the first thing that we want to do is make sure we only grab](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=153) [rows that have our data in, so we want to exclude things like header and footer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=158) [So, we're just going to access the GridView Row event arg, which is e, and we're going to look for the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=162) [Row.RowType and if it is not equal to a DataControlRowType.DataRow, we are going to just return back out of that event.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=168) [Okay, so that's fine. Now let's go through and declare a hyperlink and that is actually going to be cast to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=180) [a control type of hyperlink and where we're going to get that from is e.Row.Cells 0, so it's going to be the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=188) [first cell, and we're going to grab the first control in that cell.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=196) [Now what we want to do is set the text attribute of that hyperlink to an HTML encoded version.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=201) [So, we're going to grab AntiXssEncoder again and we're going to HTML encode the hyperlink.Text and we can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=210) [useNamedEntities again as well, that is just fine. So let's rebuild the app and now we'll jump back into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=220) [the browser and give it another go. Okay, let's give the page a Refresh and here we go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=226) [So, now rather than that alert actually being rendered into the markup and causing an actual alert, which then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=235) [appears in the browser like it did before, it has now HTML encoded the entire thing, which is why we see it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=242) [on the screen and not in the markup. So, there's a little bit of manual work there, but particularly with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=248) [Web Forms we need to remember that even persistent data, data already in our system, could contain a malicious payload.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=254) [Now where could this data come from? Well, it could come from a compromised admin interface that didn't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=263) [implement whitelisting or requestValidation or any of the other things that we've looked at earlier on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=268) [It may have come from a SQL injection vulnerability. It may have even come by migrating data from another system,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=274) [which had other risks. There are many different possible options and it doesn't really matter where it came from,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=280) [the point is that we must expect that Persistent XSS can exist and that means always encoding your output for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=287) [the correct context in which it's rendered.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=11&mode=live&start=295)

[Demo: Native browser defences](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live)

[So far throughout this module, I've been using Firefox the whole time](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=0) [and I'd like to demonstrate now why that is because there are some very important differences between the way](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=4) [Firefox and Internet Explorer, as well as other browsers handle cross site scripting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=11) [Let me paste in one of the examples we used earlier, which was the JavaScript CookieStealer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=17) [Now this was the one that embedded a script tag and accessed the cookies so let's press Enter on that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=24) [and you will see that this does exactly what it does before. I've reverted the web application to an insecure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=31) [state in order to demonstrate this. Alright, let's say I coded that and now what I'd like to do is go and show](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=37) [you what happens in Internet Explorer. Okay, so now we're back in IE10 and I'm going to paste in that XSS string](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=43) [with the JavaScript again and we're going to see something quite different and here it is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=53) [IE10 has given us a little popup saying Internet Explorer has modified this page to help prevent cross site scripting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=58) [Now this is nothing that we have implemented in the XSS mitigations.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=66) [This is a native browser defense that IE implements that Firefox didn't.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=71) [Now this is important because it shows that different browsers have their own defenses and their own ways of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=77) [approaching potentially malicious XSS. Now, of course, this can also be problematic too because it may lead to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=83) [false positives depending on the website and this is actually something that we can customize for Internet Explorer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=91) [So let's take a look at how that works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=98) [So, I'm back in Visual Studio and I'm in the Web.config and what I want to do is just add a section to add](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=102) [a customHeader. I'm going to cheat a little bit this time and paste from my clipboard a section for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=110) [system.webServer and we're going to add a customHeader that is called X-XSS-Protection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=117) [Now this is a nonstandard header, it's not one from the HTTP spec, hence it's beginning with X.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=125) [Let's go back to Internet Explorer and have a look at what this looks like after saving the file.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=131) [Alright, so we're back in IE, we've still got the page loaded where it natively defended against that script](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=139) [in the URL before. Now let's reload the page and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=145) [Okay, so we have ended up at evilsite and we have ended up with the JavaScript prompt again.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=151) [So, we've actually been able to disable it with that header. Let's just confirm that that is indeed what has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=156) [changed the behavior. So, what I'm going to do is dismiss that and we are going to go back to XSS and let's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=162) [open up our Developer Tools with F12, jump over to Network, Start Capturing, and reload the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=172) [We'll double-click to drill down on that first request and we'll jump over to the Response headers, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=184) [we see that we have an X-XSS-Protection Response Header again and it is set to 0, which effectively turns it off.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=191) [So, there are two important things to take away from this particular exercise.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=199) [One is that browsers do provide some form of XSS defense, it differs between browsers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=204)[Chrome has a very similar behavior to Internet Explorer, but it doesn't tell you, it just stops the XSS from](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=211) [executing. Firefox didn't catch that one at all. So, that's the first thing.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=217) [The second thing is, is that we can control the behavior of the XSS defense through custom headers such as the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=221) [one that we're looking at here. Most importantly though, we should never just rely on the browser implementation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=228) [of an XSS defense to protect our users. Now this is just the same as not relying on request validation to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=235) [protect our users rather than actually implementing robust and comprehensive defenses ourselves by doing things](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=244) [like whitelist validation and output encoding. So, again, we're back to this concept of security in depth,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=252) [where we apply mitigations in layers upon layers upon layers and the native browser defenses are just one of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=258) [those layers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=12&mode=live&start=265)

[Demo: Payload obfuscation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live)

[In order for a reflected XSS attack to be successful, it's entirely dependent on having victims](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=0) [follow a malicious link, which carries the XSS payload. Now in order to do that, an attacker needs to be able](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=8) [to convince the user to follow a link. So, what they'll do is go to some links to try and obfuscate the intent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=15) [of the link. Many users probably wouldn't notice a script tag in a URL anyway, but there are various approaches](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=22) [that an attacker can take to ensure that they can't even see it to begin with.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=30) [One of those approaches is to URL-encode every single character. So I'd like to take a look at how that looks.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=34) [So let's do our classic search for lager and in my clipboard, I have a URL-encoded string and I'm just going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=41) [to replace the search term and paste that in. So, have a look at this.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=51) [This is every single character URL-encoded into a hex value. Now what that means is that you cannot tell](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=56) [just by looking at this, what the intent of that URL is. Let's load it and see what happens, and here we go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=65) [It's the same JavaScript cookie stealing attack that we've seen many times now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=73) [Now, of course, we had no way of knowing what was in there because the URL was obfuscated and that's one very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=78) [effective approach an attacker can use. Let's have a look at another approach, which uses URL shorteners.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=85) [So, I'm going to dismiss this alert and I am going to go and copy another URL into my clipboard.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=93) [Now you'll notice that this URL is on the tinyurl domain. Now tinyurl, like other tools such as bitly,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=103) [is a URL shortener. So behind this URL is another URL. Now we don't know where it's going to be until](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=111) [we actually load it up. So, let's hit that and it's the same address again, it's the same XSS payload.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=118) [This time, though, we've managed to obfuscate it by turning that into a tiny URL and obviously any number](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=128)[of different URL shorteners can be used to achieve the same effect.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=134) [So this is another tool in the hacker's arsenal in order for them to keep a malicious URL out of sight of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=139) [an unsuspecting user. You would have to be extremely diligent to know what was going to happen behind that URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=146) [before you actually loaded it into the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=13&mode=live&start=153)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live)

[We've covered a lot of material in this module, but there's a few key things to keep in mind when you're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=0) [protecting your application from XSS and the first most important point is output encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=5) [This is the cornerstone of XSS protection and it's very important you remember to output encode for the correct context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=13) [Output encoding for HTML is different to JavaScript is different to CSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=21) [Even within HTML, output encoding HTML attributes is different to output encoding into the body of HTML.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=26) [And speaking of output encoding, be very wary about the way it's implemented in web forms.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=34)[Different controls have very different approaches to native encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=41) [Even the same control may implement output encoding on some properties, but not on other properties, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=46) [we saw that on the tool tip of the label control. ASP.NET MVC does a great job of offering implicit coding support.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=52) [In fact, if you don't want to enable output encoding, you have to use that Html.Raw helper.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=60) [So it's a very explicit process when you don't want output encoding on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=66) [Now, of course, just like in the previous module on injection, whitelists are also still very important.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=71) [This is almost a mantra -- Always validate all untrusted data against a whitelist of allowable values.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=78) [So, that's a very important layer of defense against XSS attacks.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=84) [Request validation is a fantastic feature in ASP.NET. Don't turn it off, but certainly leverage the ability,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=89) [particularly of ASP.NET 4.5, to fine-tune the individual fields that you want to validate vs. the ones that you don't,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=97) [but it is a safety net, don't rely on it. It should only be there in addition to all of your other good XSS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=107) [mitigations. Code as though it doesn't exist. You also need to code as though you've already got XSS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=114) [in your system. You could have a database full of persistent XSS so don't always just expect XSS to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=122) [reflected from the query string. In terms of browser defenses, there are XSS implementations to try and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=129)[protect users from sites that haven't been able to protect them themselves that don't rely on them.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=136) [Again, it's a safety net and they are inconsistent between different browsers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=142) [That will also continue to evolve so expect to see changes in further versions of Firefox, as well as all](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=147) [the other major browsers. And finally, attackers will try every sneaky trick they can to get XSS payloads](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=153) [through to an unsuspecting victim. So, expect obfuscated URLs with lots of encoding and also expect URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=161) [shorteners, which are attempting to hide the true intent of the target URL with an XSS payload.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m2-xss&clip=14&mode=live&start=168)

[Broken Authentication and Session Management](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live)

[This is Troy Hunt and welcome to Part 3 of the OWASP Top 10 on Broken Authentication and Session Management.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=0) [In this module, we're going to start out by taking a look at how OWASP actually views the risk and we're then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=8) [going to move on and actually perform an attack on a vulnerable application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=14) [We are going to exploit the risk and we are going to do some session hijacking.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=17) [I'd then like to talk a little bit more about what it means to persist sessions across http and we'll have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=23) [a look at some of the secure configuration that we can do within ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=30) [We'll also take a look at using the native authentication and membership features that are available in ASP.NET](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=36) [and that addresses that membership component of this risk. We're then going to move on and have a look at timeouts](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=41) [as well. There are a couple of different ways that we can configure timeouts to be more secure within ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=0&mode=live&start=48)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live)

[So, when OWASP defines the Threat Agents, they're really talking about pretty much anyone.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=0) [It could be external anonymous users, it could be people who already have an account.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=4) [Those Threat Agents could come from literally anywhere.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=9) [The Attack Vector typically exploits the accessibility of session IDs.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=13) [When session IDs can be exposed, that's when we get things like session hijacking or session fixation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=19) [The Prevalence is pretty common and one of the reasons it's pretty common is that we often see developers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=27) [building their own session management or authentication modules and it's very easy to introduce flaws into those.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=32) [So one of the things we're going to look at in this module is leveraging the native functionality rather than](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=38) [trying to roll your own.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=43) [The impact of broken authentication and session management can be severe and the reason it's so severe is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=46) [because when it happens, it typically leads to someone's account being hijacked; it allows an attacker to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=52) [actually impersonate the victim and then gain access to whatever they may be able to gain access to while](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=57) [still authenticated to the application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=63) [And finally, the Business Impact is obviously going to be dependent on the nature of the application and the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=65) [nature of the data that has been exposed to an attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=1&mode=live&start=69)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live)

[What I'd like to do now is actually go through and exploit an application that is at risk of broken authentication](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=0) [and session management. I have an app that is running on my local machine and it has a host entry called brokenauth.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=6) [Now this is just an ASP.NET MVC 4 application, obviously just a standard template, and this app is prompting me](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=15) [to log in and I am going to log in as troy and I'm going to put my Password as hopefully something very secure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=23)[Now what you'll notice just before I submit this form is that in the URL, we had a string appended to domain](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=32) [and this string is actually the session ID. Now what we're going to see here is the session ID being persisted](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=40) [in the URL and why that is risky. So let's log on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=47) [Now what happens here is it comes back and tells me that I'm logged in as troy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=53) [If I navigate through to other parts of the site and then I come back to the homepage, I'm still logged in as troy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=57) [So, what's happening here is obviously there's a piece of information about me and my session, which is being](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=65) [persisted between requests. That's how it still knows that I'm troy when I get back to the homepage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=70) [Now here's the risk. What we're going to do is take this URL, Copy it, drag over a Google Chrome window](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=77) [that also has the same website open, and you're going to see a couple of things here that are a little bit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=83) [different and the first thing that's a bit different is that the path in the URL is different.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=87) [So, what you'll notice is that the session ID, which is all of this piece after the domain, is different to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=94) [the one in Internet Explorer. As a result, I'm not logged in, I'm still seeing the Log in form.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=100) [If I refresh that, we still see the Log in form. Now what I want to do is, before I copied the URL out of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=105) [Internet Explorer, let's now paste that URL into Chrome and, of course, it's going to have the same session ID.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=113) [So, what we're now seeing is that Chrome has been logged in as myself.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=120) [So, effectively what we've done is we have hijacked the session from Internet Explorer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=124) [So, all it would take is that URL in Internet Explorer to be shared with somebody else and the session can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=130) [be hijacked. How could it be shared? Well it could be shared via social media or it could be sent in an email,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=135) [it could be a victim very innocently just saying, "hey I want to send a link to a website because I think you](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=143) [might be interested in it" without knowing that it's actually going to give somebody else control of their session.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=2&mode=live&start=148)

[Risk in practice: Apple's session fixation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live)

[Now to try and give some sense of how this might actually happen in the world and show that it's really not](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=1) [just a hypothetical, there was a case a little while ago where someone found that whilst configuring](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=5) [Mountain Lion for our sakes, there were session IDs that were leaked in URLs that could be observed across](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=11) [a wireless network and, in fact in this case, what was found was that if those session IDs could be accessed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=18) [they actually had the ability to go through and do things like change the verified email address and the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=26) [security settings of the Apple account, which is obviously a fairly serious situation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=30) [Now in this case, what was found is that once the user logged in, the session ID was persisted in the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=36) [So, there wasn't necessarily even packet sniffing or looking at request headers or other things that we'll](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=44) [talk about in later modules; this was a very simple case of the URL being leaked and the URL having sensitive](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=49) [data in it. Now this is one of the reasons why we always say, never have any sensitive data in the URL,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=56) [even over HTTPS requests.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=3&mode=live&start=63)

[Persisting state in a stateless protocol](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live)

[So, we've seen session IDs in the URL. Let's actually take a little step back and talk about why we need](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=0) [session IDs at all and it all comes back to HTTP being a stateless protocol.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=7) [So, what we mean by that is that when someone makes a request to a website, there is nothing that keeps the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=13) [connection open and keeps the identity of the user or the identity of the session persisted across requests.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=19) [In order to uniquely and securely identify someone across those requests, we need some sort of manual construct](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=27) [and the manual construct that we used in the examples earlier is we had a session ID in the URL, so every time](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=36) [the request was made, it passed that little piece of information back, and that's how we sort of hack around](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=43) [the inability of HTTP to natively identify that subsequent requests come from the same person in the same](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=49) [authenticated session.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=4&mode=live&start=56)

[The risk of session persistence in the URL versus cookies](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live)

[So let's go back to session persistence via the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=0) [So, what we saw before was an ID being uniquely generated and then that ID being inserted in the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=3.5) [As we navigated around the site, that same ID was persisted in each request so the web server was able to say,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=11.5) [hey this is the same person because they're coming in with the same ID.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=18.5) [Now ASP.NET did all of that natively under the covers and it was all very automated.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=22.5) [Now the problem with doing this, as we just saw, is that if somebody else gets that URL then we have a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=29.5) [session hijacking risk. Now they can get that URL in many different ways, so the easy ones would be someone](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=35.5) [simply sharing it, "hey I saw something interesting on a website that I was authenticated to, here's the link."](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=43.5) [They might Tweet it, they might put it on Facebook. When they do that, if the link contains the session ID](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=50.5) [and the session is still active, anyone that clicks on it will hijack the session.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=55.5) [URLs are also very often logged at various points in the network communication so they may be logged via](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=60.5)[proxy server as the request goes out, they may be logged in the web server logs, and even though the web server](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=66.5) [logs should be private, that's enough information for people to start hijacking the session if anyone does](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=72.5) [gain access to those web server logs. The other problem, of course, is that URLs sit in your browser history.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=78.5) [So, you may be finished with a site, walk away, and somebody else comes over and looks in the browser history,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=85.5) [and even if you've closed the browser, they may be able to find that site and, again, if the session is still](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=91.5) [active, hijack the session. So URLs are really a very, very insecure place to store anything sensitive and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=95.5) [the session ID is sensitive.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=103.5) [So, the alternative to persisting the session ID in the URL is persisting it in a cookie.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=107.5) [Now this starts out the same way in that the web server will generate a unique identifier, but rather than](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=113.5) [putting it in the URL, it will actually put it in cookies. So, it will send it as part of the response header](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=119.5) [after somebody authenticates or after the session begins. Now, what that means is that when the browser](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=125.5) [subsequently makes requests to the website, those requests will include any cookies of that domain.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=132.5) [So that's the way we can pass this little piece of information back to the web server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=138.5) [It won't appear in the URL, it will simply be saved in the cookie. So now, unless somebody can access the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=143.5) [request headers and we'll talk about this in subsequent modules, nobody can hijack the session, they need to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=150.5) [get the information in the request header. The only real downside to using cookies is that you need to have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=155.5) [cookies enabled in the browser. For the most part, though, just about everybody does.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=163.5) [It's a rare thing to find that cookies aren't enabled in the browser and if they aren't, it really does create](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=169.5) [a pretty serious security risk when we talk about how we persist sessions.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=5&mode=live&start=175.5)

[Demo: Securely configuring session persistence](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live)

[Now in ASP.NET, we've actually got four different ways we can configure session persistence and the first way](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=0) [is what we've already seen in action in the browser using the URI. So, we'll take a look at our Web.config](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=7.5) [in a moment and what you'll see is that it was configured to explicitly persist the session via the URI.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=13.5) [Now the other option, which we've also just looked at, is using cookies and using cookies is the default position](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=20.5) [of ASP.NET. Now this means it's always going to use cookies, whether the browser supports it or not.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=27.5) [So by extension of that, the only way that you can persist the session or persist authentication is if you have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=33.5) [cookies enabled in your browser. There are two other options, though, and one of those options is to use the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=38.5) [device profile and this is ASP.NET saying, "Does this particular browser have the capability to support cookies"](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=44.5) [and if it does, it's going to use cookies and if it doesn't, it's going to persist the session ID in the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=53.5) [The risk with that one is that if the browser does support cookies, but somebody has disabled them, then it's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=61.5) [not going to work. The final position, which is a little bit sort of in-between all the other ones, is to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=68.5) [AutoDetect and ASP.NET will actually detect whether the browser is configured to accept cookies.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=74.5) [So it sounds a little bit like the best of both worlds. If it can support cookies, it will use them;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=82.5) [if it can't, it will use the URI. Of course, you do have to ask whether it's ever secure to use the URI and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=87.5) [perhaps if someone doesn't have cookies enabled, it may just be better to not even grant them access as opposed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=94.5) [to creating an insecure session and that's the default position because ASP.NET is set to use cookies by default.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=100.5) [So let's go and configure this application to go back to the default and use cookies.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=108.5) [So, the reason it was using URI is configured here in the sessionState and it's simply because we've been](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=113.5) [very explicit and we've actually told it to use the URI. So what I'd like to do is actually remove that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=118.5) [definition and allow it to go back to the default. We'll save that file and now let's jump back into Internet Explorer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=124.5) [and see what happens when we start a session that doesn't actually have URI supported.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=132.5) [So let's go back to the original site and the first thing we'll see when the site loads this time is that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=138.5) [do not have a session in the address bar anymore, there is no session ID.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=144.5) [So let's try loading it again and we get the same behavior. It tells me I'm logged in as troy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=149.5) [If we go to About, go to Contact, and then we go back Home again, I'm still logged in as troy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=159.5) [So, my session is still being persisted, but clearly it's not being persisted via the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=165.5) [Let's open up the Developer Tools by hitting F12 and we'll go over to Network and let's capture one of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=173.5) [requests that we make from the browser. So, we'll just reload the page and here is the request to the homepage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=181.5) [Let's open that up and have a look at the cookies and now what we see is the session ID actually stored in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=192.5) [the cookie. So it's exactly the same experience to the end-user, it's just that now it's not in the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=199.5) [Now, of course, if we copy and paste that URL into another browser or send it to anybody else, it's just going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=205.5) [to start a brand-new session because there's information in there that persists my session.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=6&mode=live&start=210.5)

[Demo: Leveraging ASP.NET membership provider for authentication](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live)

[So far we've been very focused on session management, but of course this particular risk in the OWASP Top 10](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=0) [is about broken authentication and session management. Now whilst the two are closely related, there are some](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=7.5) [very important differences and, of course, one very important thing about authentication is providing credentials](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=14.5) [and actually logging the user on and then having the ability to do things like authorization.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=20.5) [OWASP talks about avoiding rolling your own authentication schemes and what they're talking about is saying,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=27.5) ["Look authentication can be a complex process." If it's possible to do it with the facilities within the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=35.5) [framework, use something that's tried and tested and has been very well proven.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=41.5) [So what we're going to do is just start a brand-new ASP.NET MVC 4 web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=47.5) [It's going to be an internet application because we are going to want to use forms authentication and forms](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=54.5) [authentication is where we're going to look at how we might be able to actually create facilities, such as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=60.5) [registration, and then provide the ability to do things like log on to the application and all of this is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=66.5) [available within the ASP.NET membership provider. The only thing we need to do to make this work is to set](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=73.5) [the connectionString in the Web.config and we'll see that by default, we have a connectionString in there.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=81.5) [I'm going to set that to a version that will point to my local SQL Server, it's going to use a database called](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=87.5) [3-BrokenAuth, and it's going to use an account, which I've already created.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=98.5) [Now before we go any further, I just want to take a quick look at what is actually in that SQL Server database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=107.5) [Now if we have a look at 3-BrokenAuth and we drill down into the tables, we'll see that there are no tables](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=115.5) [at all in the database; it's a brand-new database. All that's happened is I've added that user account and that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=123.5) [user account has database owner rights. Let's now flip back to the application and we're just going to run it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=130.5) [So, obviously this is the first run of a brand-new application. Nothing has been configured other than the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=147.5) [connectionString and we're going to register and I will create a User name of troy and for the purposes of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=153.5) [this demo, it will be a very basic Password.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=162.5) [And that's it and now we're logged on as troy, we have facilities such as Log off, and then we have the ability](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=169.5) [to log back in again and we'll just validate that that feature does indeed work and that's great.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=176.5) [So, with nothing more than the configuration of a connectionString, we now have the ability to register, log off,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=185.5) [and log on. Let's take a quick look at what's happened in the SQL database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=192.5) [What we can now see is that we have five new tables that have been created.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=202.5)[These were automatically created when we first ran the application and invoked the membership provider by](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=207.5) [registering. If we have a look in the Membership table, we should find one record for me.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=214.5) [This is my current time and here is an account with a securely hashed password, which we'll talk about later on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=223.5) [when we get to cryptographic storage. So all of that has been created automatically for me.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=233.5) [Again, it's a very well tried and tested model, the ASP.NET membership provider, and it keeps part 3 of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=238.5) [OWASP Top 10 very happy because we haven't gone and built it from scratch and possibly introduced new risks.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=7&mode=live&start=245.5)

[Customising session and forms timeouts to minimise risk windows](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live)

[So, another thing OWASP talks about in broken authentication and session management is timeouts and timeouts](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=0) [are all about when the session should be automatically expired and this is important because you don't want to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=7) [have sessions just go on for perpetuity that is a security risk. There are some cases where we may need that,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=13) [but for the most part, it's not a desirable place to be. So when we talk about sessions, let's imagine that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=20) [we have a point where a session begins and we have a point where a session ends.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=28) [How it ends is not too important for the purposes of this slide. What I want to talk about is during this time,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=33) [obviously there is a convenience to the user. So, while someone's session is persisted, whilst they're still](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=41) [authenticated, they can move around the site as an authenticated user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=47) [They might be able to walk away and get a cup of tea or have some lunch and come back and they're still](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=53) [authenticated so they don't have to log on again. That's a nice convenience for a user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=56) [The flipside, of course, is that whilst someone is authenticated, their session may be hijacked.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=64) [There is a risk of session hijacking either through the means that we just looked at or via other means as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=70) [So, for example, sniffing packets on an unsecured wireless network and we'll talk about that a little bit more](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=77) [later on. The point here is that there is a tradeoff between the convenience and the security and it's our job](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=82) [to try and find a happy balance in-between those two.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=91) [So let's look at the timeouts for both session and forms and they are both a little bit different so remember](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=95) [forms is about being authenticated, sessions may be just simply persisting a little piece of data across requests.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=101) [Now the default timeout of a session is 20 minutes. If we put something into session state, which is what we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=109) [did in that very first example with the session hijacking, after 20 minutes of inactivity the session will end](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=115) [and the data will be discarded. Forms authentication is 30 minutes, so by default forms authentication will](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=123) [log somebody out after 30 minutes of inactivity. Now there is a little issue though because Visual Studio,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=132) [when you create a brand-new ASP.NET web application, it massively increases the forms timeout.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=142) [So, instead of being 30 minutes, it changes it to nearly 3,000 minutes, which is actually 2 days.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=148) [So what we need to remember is if we create a brand-new ASP.NET web application and we use forms authentication,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=155) [your timeout is actually going to be very long and you need to have this discussion about -- is the convenience](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=162) [that that brings worth the risk that it also brings. Inevitably, defaults are not the perfect value for every](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=169) [situation, perhaps it should be 1 day, perhaps it should be 1 hour.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=177) [That's a very, very case by case specific discussion.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=8&mode=live&start=182)

[Siding versus fixed forms timeout](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live)

[Now one of the things we can do with forms authentication timeouts is we can configure what's referred to as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=0) [sliding expiration. We can't actually do this with session timeouts, it's only for forms.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=6) [So let's take a look at what we mean by sliding expiration. So let's imagine we have an ASP.NET application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=12.5) [that has forms authentication enabled and it has the default 30 minute forms timeout.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=21) [So let's say the session starts at 9:00 a.m. Now by default, that session is going to end at 9:30.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=28) [Now what will change that in a sliding expiration timeout world is if there's another request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=38) [So, if the user makes a request at 9:10, then the timeout starts again, and this is why it's called sliding,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=45) [because it will just keep extending. So, what this means is, is that the session will only ever time out](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=52) [30 minutes after the last request. So, there's always sort of this period of grace just in case the person](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=60) [comes back and wants to keep using the application. Now, again, that's a convenience thing because 30 minutes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=65) [gives you time to go and get a cup of tea or chat to a colleague. A nice little convenient feature, but of course](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=72) [it's a security risk in that it does give us a greater period of time where the session can be hijacked.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=80) [Now the other option is to apply a fixed timeout, so rather than letting it slide, we're going to be explicit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=87) [about when the timeout finishes. So let's take the same scenario -- the timeout duration is 30 minutes, which](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=94) [means the session starts at 9:00 and the same as before, it's going to end at 9:30.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=102) [Now before somebody made a request that pushed the timeout back, but when we fix the timeout and we don't use](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=108) [sliding expiration, it doesn't matter how many requests you make, that session is still going to end at the end](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=114) [of the session timeout. Now, obviously the downside to this is that if somebody is actively using the application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=120) [for longer than the session timeout duration, so for longer than 30 minutes in this case, they're going to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=130) [logged out while they're in the middle of doing something. The flipside is, is that from a security perspective,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=135) [the window of opportunity for this session to be hijacked is smaller.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=144) [Clearly there's a tradeoff between usability and security and in many cases, a sliding expiration won't make](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=150) [a lot of sense, but it is there as an option.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=155) [Now sliding expiration is very easy to configure and by default, forms authentication will default to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=161) [sliding expiration being on, so every request will keep extending when the session actually expires.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=168) [Now we can very easily set it to being fixed simply by turning it off on the forms authentication element in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=175) [the Web.config. Be cautious with this, it may increase security, but again there is an adverse usability impact.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=181) [It may make more sense simply to change the duration of the session timeout and leave expiration as sliding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=9&mode=live&start=190)

[Other broken authentication patterns](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live)

[One of the things about this third risk in the OWASP Top 10 is it is rather broad, it's a little bit generic.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=0) [We've looked at a few different areas of authentication session management, but there are other areas that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=7) [OWASP talks about as well. So, for example, the secure cryptographic storage of credentials.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=12) [Now obviously this is important so that if your database gets breached, there's a higher barrier to entry to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=18) [actually gain access to the credentials, but it's also important because we know that people have a great](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=24) [propensity to reuse passwords across sites and there are many examples of where a website is being breached,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=30) [passwords exposed due to insecure cryptographic storage, and then other accounts the users had subsequently](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=37)[being breached. So you're not just looking after the credentials on your site, unfortunately you're looking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=43) [after the credentials on other people's sites as well. Implementing a robust minimum password criteria,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=48) [so trying to have a long minimum length and, again, there's a balance between usability and security, as well as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=55) [requiring a mix of lowercase/uppercase characters and symbols and the membership provider makes this extremely](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=61) [simple because it's all configuration-based in the Web.config. Never sending a password via email.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=69) [Email is a very insecure mechanism in transit because it's not encrypted and it's also not intended to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=77) [a secure storage facility as well. So if you do need to help somebody, for example, log back in if they've](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=83)[forgotten their password, you need to implement a secure reset process and not just pull the password out of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=91) [the database and send it via email. Also, even if we put session IDs in cookies, it doesn't make it 100% secure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=97) [We still need to be cautious about the way we protect those cookies.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=106) [So back in Part 2, we looked at cross site scripting, making sure we don't have cross site scripting](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=110) [vulnerabilities that may be able to gain access to those cookies, that would be a broken authentication risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=115) [The other thing to remember about session IDs and cookies is that we need to protect them when they're in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=122)[transit across the network. So, plain old HTTP is vulnerable to interception, particularly somewhere like a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=127) [wireless hotspot. So when we get to Part 9 of the OWASP Top 10, we'll look at insufficient transport layer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=134) [protection and we'll look at the risk of cookies being sniffed and sessions being hijacked.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=140) [So, we need to keep that one in mind as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=10&mode=live&start=145)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live)

[So, in Summary, whatever you do, keep session IDs out of the URL, put them in cookies, and of course as we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=0) [just looked at in the last part, make sure that cookies are properly protected as well from risks such as XSS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=6.5) [and insufficient transport layer protection. Use that ASP.NET membership provider wherever you can.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=13) [A lot of work has gone into it, it's been extensively tested, and it takes a lot of work away from you as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=20) [a developer as well. It also makes it very easy to do things like define minimum password criteria.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=25) [It's a fantastic starting point for building authentication. Also think about those session forms timeouts.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=31) [The default values are almost certainly not the best values for your application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=38) [So, consider the convenience of having a longer timeout so that people don't get logged out too early with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=44) [the increased security of not leaving that session vulnerable for too long.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=50) [Try and tailor those timeouts to meet your specific requirements and it's going to differ app by app.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=55) [Also consider sliding expiration. If you can disable it, it will increase the security because it will reduce](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=62) [that window of opportunity for an attacker to hijack a session, but of course it does have an adverse usability](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=69) [impact and you need to expect that sometimes people will get locked out when they weren't expecting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=75) [And finally, remember that broken authentication session management is a very broad risk and it touches on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=81) [many different aspects of software security. It touches on things we've already looked at, such as cross site](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=87) [scripting, and it will also touch on areas that we'll look at in later modules, such as insecure cryptographic](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=93) [storage and insufficient transport layer protection. It's a very good reminder that security is about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=98) [applying defense in depth over many different areas of the web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m3-broken-auth&clip=11&mode=live&start=105)

[Insecure Direct Object References](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to Part 4 of the OWASP Top 10 on Insecure Direct Object References.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live&start=0) [In this module we'll start out by taking a look at how OWASP views the risk and then we'll move on to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live&start=8) [actually performing an attack against an application at risk of insecure direct object references.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live&start=12) [In order to understand the risk a little bit better, we'll then take a look at what a direct object reference is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live&start=19) [before we move on to implementing access controls to begin mitigating the risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live&start=24) [We'll also build an indirect reference map in order to provide some abstraction between the references that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live&start=29) [we expose externally and the references that we store internally.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live&start=35) [And finally, we'll look at a means of object reference obfuscation by using surrogate keys.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=0&mode=live&start=40)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live)

[So, we're moving on to the OWASP overview and risk rating and in this case, the Threat Agents can be any users who](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=0) [might have partial access to parts of the system data and obviously that could be quite a broad range of people.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=6) [The Attack Vector is easily exploited and often it's an authorized user who's simply tampering with parameters](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=13.5) [to gain access to records, which they logically shouldn't be authorized for, but a flaw in the system allows](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=21)[them to gain access to.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=27) [The Security Weakness is prevalent, but detectability is also pretty easy and, in most cases, it simply boils down](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=29) [to the fact that there are insufficient access controls and internal identifiers are exposed externally and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=36) [it's simply a matter of an attacker manipulating those external references and gaining access to internal records.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=44) [The Technical Impact is moderate and the amount of data an attacker can gain access to is really only limited by](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=52) [the number of records they could possibly gain access to by manipulating parameters in the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=58) [As always, the Business Impact is very dependent on the nature of the data so it's going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=65) [differ significantly from case to case.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=1&mode=live&start=70)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live)

[Let's take a look at a vulnerable app and see how we can actually exploit insecure direct object references.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=0) [So, this application is telling me that I should probably try and log in to view my profile.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=7) [At the moment, I am definitely logged out, but if I go and log in and with the User name of troy and a Password](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=12) [like so, after logging in I will be taken back to the front page, and it is now recognizing that I am logged in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=22) [I have a prompt here to view my profile so let's have a look at what I get when I view my profile.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=31) [Okay and here is my data and I assure you this is dummy data and this has been loaded asynchronously.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=39) [So, there is a service behind that ‘View my profile' button, which has gone to the server and pulled my data back.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=46) [Now what we want to do is see if that service may be vulnerable to insecure direct object references.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=54) [Okay, so now that we've got Fiddler open, let's flip back to Internet Explorer and reload that profile.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=62) [And we'll just give it a quick ‘View my profile' again and now let's jump back to Fiddler and see what we see.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=68) [Okay, so here's what's happened. Fiddler now has a record in our list of HTTP requests and what we can see here](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=79) [is that we are making a request to the host name of DirectObjRef and to a URL called UserProfile and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=86) [there's a query string -- userName=troy. So, what we can conclude from this is that this API is being called by](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=94) [passing my username to it. Now what I want to do to investigate the risk of insecure direct object reference](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=103) [is start composing some new requests and manipulating that UserName parameter.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=110) [So let's drag the request over here into the composer and what we can see here now in the Request Headers is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=115) [that the entire request has been recreated and it's ready for me to execute and issue back to the server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=122) [So, if I hit the Execute button, it's going to go back and run that query again and we should get exactly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=128) [the same result back and, in fact, if we scroll this over a little bit, we should see that the Response Body](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=133) [is identical. Now I'm just going to stop capturing so that it doesn't capture any other HTTP requests made by](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=139) [my machine, such as from browsers or anything else talking over the internet, and now what I want to start](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=146) [doing is manipulating this parameter a little bit to see if we can pull any other records back.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=152) [So, let's take an example. Let's maybe search for userName=bill and we will run that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=159) [Now this time we've had an HTTP 500 return. So there's actually been an internal server so it doesn't look like](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=170) [we've been able to pull anything back from bill. Let's try it again and this time we might try something like](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=176) [mary and it's a similar result so it looks like it's not very happy with mary.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=183) [Let's try one more and we will try john and execute john and with john, we have had an HTTP 200 success.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=190) [Now, that's great. So what that's shown is that we've been able to access this API and it looks like we've been](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=202) [able to pull back somebody else's record. Let's go and take a look at what that record actually looks like and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=208) [we're going to jump over to the Inspector and the Inspector is broken up into two parts.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=214) [So, in the top part we have the Request Headers and then in the bottom part, we have everything that is in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=220) [the response. So, for example, we have our Response Headers so we can see that we've had an HTTP 200 return.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=226) [What I'm interested in though is I want to have a look at the Raw response and this Raw response is going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=233) [show us the Response Body, as well as all the headers, but it's the Response Body that I'm most interested in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=239) [Now, it looks like we've had a JSON response here so I'm going to go over and hit the JSON tab and now we have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=245) [a very nicely formatted view of John's record and, in fact, what we've been able to do is gain access to john's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=251) [Address, his BirthDate, and his TaxFileNumber. So, this is a very serious case of exploiting application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=259) [at risk of insecure direct object reference. Now the direct object reference, of course, is the UserName.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=267) [We've simply enumerated through some logical possibilities of what the UserName might be and after a little while,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=274) [we've hit on one that is valid. So, clearly there is something wrong in that this application is allowing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=280) [this to happen. Okay, so now that we understand that, let's go and have a look at what a direct object reference](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=286) [is and then how we can mitigate this risk using access controls and an indirect object reference map.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=2&mode=live&start=292)

[Risk in practice: Citibank](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live)

[As usual, let's take a look at an industry precedent where insecure direct object references have been exploited](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=0.5) [to do some serious damage. Now in this case, in 2011, City Bank had an insecure direct object reference flaw,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=7.5) [which was responsible for exposing 200,000 customers' details.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=16.5) [Now this is a very serious flaw because we're talking about a bank here that manages people's money so you](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=23.5) [wouldn't expect this sort of flaw to occur on that sort of website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=28.5) [Now what's interesting here is when you look at the way the attack was executed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=33.5) [So, they talk about the attackers being able to gain access to data by changing the numbers in the URLs that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=38.5) [appeared after customers had entered valid usernames and passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=44.5) [So, this is one thing about the insecure direct object reference flaw, it's often an authenticated user who](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=49.5) [can gain access to the records; it's just that they're gaining access to other people's records.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=56.5) [And as is pointed out here on the register's website, this was indeed a case of insecure direct object reference.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=61.5) [Now in this case, it looks like the attackers were simply changing account numbers, which is an extremely simple](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=68.5) [risk to exploit; they're just simply changing values in the URL and obviously there haven't been sufficient](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=75.5) [access controls to ensure that the person logged in is actually allowed to access the records that they're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=81.5) [trying to request. So, a very good example of insecure direct object references and a particular good example](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=88.5) [given it's a bank that has demonstrated the risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=3&mode=live&start=94.5)

[Understanding direct object references](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live)

[Let's take a closer look at how an insecure direct object reference leaves a web application at risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=0) [So the first thing to understand is that when we talk about an object reference, we're normally talking about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=9.5) [some sort of a key or identifier and in this case, it's one that is publicly observable or at least observable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=14.5) [to the attacker and, again, that attacker may be authenticated as another user; they're just trying to gain](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=21.5) [access to records which they shouldn't be getting to. So let's assume that that ID they're seeing points to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=27.5) [a database record. In our earlier example, it was a UserName and clearly the UserName troy pointed to a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=33.5) [database record that belonged to me. Let's now go through the sequence of how direct object reference is exposed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=40.5)[So, imagine a customer logs onto their bank and then loads the page, which has a list of their accounts.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=49.5) [The bank's website then responds and provides a link to each of the accounts and the link to each account has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=55.5) [a query string and in that query string is the account number.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=62.5) [Now that account number would be a direct object reference. That particular key refers directly to a record](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=66.5) [in the database, there's no abstraction in-between it. So long as there's no access controls prohibiting the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=74.5) [customer from actually retrieving the record, they'll be able to make a request and view whatever data is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=80.5) [returned for that bank account. So here are some examples: In this particular case, we can see that the URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=85.5) [has a query string called id and what we're seeing here is naturally incrementing id's and certainly many](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=94.5) [insecure direct object references are exploited because we do have sequentially incrementing id's.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=101.5) [They adhere to a predictable pattern.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=107.5) [Now, of course, there is a risk with all of this and the real risk is that if that reference can be manipulated](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=111.5) [to refer to another record and that record retrieved without sufficient authorization, that's when the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=118.5) [direct object reference risk manifests itself. Now in order for an attacker to exploit a direct object reference,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=126.5) [they have to be able to reconstruct them. So we talked about attackers often already being authenticated and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=133.5) [simply accessing records, which they're not authorized for. Now, of course, the only way that that can happen](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=140.5) [is if they can conceive of the keys to other records. So this happens when we have patterns such as uniformly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=145.5) [incrementing integers. So, on the previous slide we saw account numbers that simply incremented.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=153.5) [Now there's nothing to stop an attacker from looking at a URL and adding one to it or adding two or three or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=159.5)[continuing to add numbers until they retrieve valid records. In the example of our insecure web application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=165.5) [earlier on, we had a natural key. So a natural key in so far as it was made up from logical data as my name.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=171.5) [So natural keys are at risk because they can also be brute forced. So we could go through and enumerate through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=179.5) [a list of names and potentially find records, which are not properly secured.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=185.5) [The other type of data that we often see in insecure direct object references is discoverable data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=190.5) [So, it may not necessarily be data which we can simply increment, but it may be data that we can retrieve from](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=196.5) [somewhere else. Now, whilst ID, such as social security numbers or tax file numbers or national IDs,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=202.5) [are normally meant to be secure in so far as they're not something that you would freely share, we do tend to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=208.5) [find that many of these numbers get exposed through all sorts of other means -- other data breaches, staff who](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=215.5) [have access to data where you might need to provide that information - and it's entirely conceivable that an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=221.5) [attacker could gain access to those numbers. Go through an application that uses them as a direct object](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=226.5) [reference and try to gain unauthorized access to the record.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=232.5) [Now when an attacker does this, they may just be sitting there at the browser and simply changing the ID in a URL,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=236.5) [but they may also automate it. It's very easy to script an attack, which simply increments a number or simply](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=243.5) [goes through a dictionary and looks for words that it can append to a query string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=250.5) [So, automation and brute force is an approach that we often see in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=255.5) [exploiting the risk of insecure direct object references.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=4&mode=live&start=259.5)

[Demo: Implementing access controls](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live)

[Now the real problem with insecure direct object references is access controls.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=0) [So when we exploited that application before and we saw the City Bank case, the real issue was that there was](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=6) [not appropriate authorization before that record was loaded. Often we see direct object reference vulnerabilities](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=12)[appear because there is an assumption that the URL is safe because it may not be immediately obvious or it may](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=20) [be hidden behind an API; it may have an ID in it, which requires tampering in order to gain access to other data,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=28) [but as we've just seen, tampering is often not very hard so certainly you can't assume that the URL is safe](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=35) [just because it's not highly visible.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=42) [Let's now take a look at the application that we managed to exploit earlier on and in this application,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=46) [I have an MVC controller called UserProfileController and we've got an ActionResult called userName.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=53) [So, this matches that path that we were making requests to in Fiddler before; the path was UserProfile and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=61) [we passed a query string called userName. Now what this code is actually doing is it is making sure that the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=67) [user is authenticated and as we discussed earlier on, insecure direct object references are often exploited](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=72) [by users who are authenticated, it's just that they're gaining access to somebody else's record.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=80) [So, in this case, if you're not authenticated, there would be an exception raised and normally it would be a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=87) [little bit more graceful than just throwing an ApplicationException, but this gives us a good idea of how that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=92) [might work. Now in this case, there is a profile being retrieved from an Entity framework data context and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=97) [that profile simply needs to match the userName that was passed in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=104) [So, obviously our risk here is that so long as you can provide a userName to the Controller action, then that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=108) [userName will be pulled back from the database. If the userName was null, there was a different exception throw,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=114) ["Profile does not exist"; otherwise, it would go through and it would return a JSON response of the Address,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=120) [the BirthDate, and the TaxFileNumber. So, what we want to do now is try and lock this thing down a little bit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=126) [and the first place to start with locking it down is to put in some form of access control.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=133) [Now ultimately at the heart of every insecure direct object reference risk is a lack of access controls.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=139) [The underlying problem in our example and the City Bank example is that authorized users are gaining access](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=147) [to somebody else's material, so there is an authorization failure or an access control failure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=155) [So let's go in and implement the proper access control. Okay, so now let's go and make sure that the logged-on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=161) [user can only access their own record and not somebody else's. So, what we're going to do is after we make sure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=167) [that they are already authenticated, let's just check that the User.Identity.Name, if it does not equal](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=174) [the userName that they're trying to access, we will throw a new ApplicationException and we will just say](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=185) ["User not authorized." Now this is a very basic access control and, in fact, you could probably speculate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=193) [that maybe it would be better to simply use User.Identity.Name when we're loading up that record anyway rather](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=201) [than allowing someone to pass the userName they'd like into the Controller action.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=208) [Where this might actually be useful though is there are valid cases where someone other than the user might](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=214) [want to access their record. So, for example, we could have an administrator wanting to gain access.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=220) [So, what we might do is something like && "User.Is(not)InRole("Admin")".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=226) [So, in that case, either you have to access your own record or an administrator has to access your record,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=236) [but you wouldn't be able to not be an administrator and access somebody else's record, but we don't need that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=242) [for now so let's just go back and we will stick with making sure that you are trying to access your own record,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=248) [rebuild it. Now let's go and jump back into Internet Explorer and make sure that it still works in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=254) [legitimate use case and then we'll try and exploit it again via Fiddler.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=259) [Alright, so back in Internet Explorer, let's go and ‘View my profile' now and see what we get.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=265) [And we still get valid data, which is great and this is my data, it is not john's data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=271) [So, the valid use case works fine. Now let's jump over to Fiddler and see if we can still exploit the app.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=277) [So in my Composer here, I still have the request that tries to gain access to john, let's just execute that,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=284) [and we get another 500, which is good, so we can't illegally access john's record.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=291)[Let's just be doubly sure that I can access troy, execute, and we've got an HTTP 200 so that works just fine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=297) [So this is a very, very simple case of just implementing the correct access controls and ultimately that's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=306) [really what insecure direct object references boil down -- have you got the right access controls to ensure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=313) [that the person trying to access the record is actually authorized to view it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=5&mode=live&start=320)

[Understanding indirect reference maps](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live)

[Access controls are really the most fundamental mitigation against insecure direct object reference risks,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=0) [but another very common way of mitigating the risk and one that we particularly see in high value applications](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=7) [protecting sensitive data, such as in bank applications, is the concept of indirect object references.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=14) [Now an indirect object reference is really a map, which tries to translate some sort of actual internal direct](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=21) [object reference to an external abstraction, which doesn't tie directly back to that internal object and can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=28) [only be used within the scope of a session by an authorized user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=35) [So think of it as a layer of defense between the user and the underlying record in the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=39) [So let's take a look at what it might look like to build an indirect object reference map.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=46) [So, we'll go back to the example from before where a customer asked the banking website for a list of their accounts](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=51)[and the banking website responds, but this time what the banking website does is it maps each internal ID to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=58) [a temporary external ID. Now that external ID is randomly generated, cryptographically random, and there's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=67) [a map between the two, which is stored on the server, so the server knows how do I map this ID, which I'm](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=75) [exposing externally to the ID, which I'm maintaining internally, and there are a few different ways that the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=82) [server can persist that and we'll look at how we're going to persist that in session ID in just a moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=88) [Now what that means is that when the customer then asks for their record, the customer is actually asking for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=94) [a record by indirect reference. It's now up to the server to translate that indirect reference to a direct](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=100)[database reference, so it's going to refer to that map that it's storing locally.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=108) [Now, of course, once the server has done this, it can respond with the actual banking record, but what this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=113) [means is that is that at no time is the internal ID exposed externally.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=118) [Because the server is only exposing cryptographically random IDs, if an attacker did try to start manipulating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=125) [that ID, it wouldn't map back to anything on the internal side.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=132) [And besides, because it's random, it doesn't comply to any sort of sequential pattern or any sort of natural key.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=136) [So an attacker would really just be guessing at the construction of the key.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=143) [Now there are a number of different ways that this map could be constructed and depending on the structure of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=149) [the application, so depending on factors such as are there multiple web front ends and maintaining session might](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=154) [be a consideration may there be a potential performance overhead with how we implement it, the design may change.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=161) [Again, the way we're going to do this is we're going to do this in session state and session state is very good](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=168) [in so far as it means the map is temporary. So once that session ends, any of those keys that have been exposed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=174) [externally are now totally useless. It's also good because the map is user-specific.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=181) [So, if one authenticated user gets access to the key that is used externally by another authenticated user,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=188) [they won't have that session map so there's no way the server can resolve that back to an internal ID of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=195)[the first user. And, of course, one of the really important things is we've got to make sure that those](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=200) [external IDs, the ones that we've generated or mapped to the internal ones, are actually random.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=205) [Now what this ends up looking like is, instead of before where we had account numbers that were numeric and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=211) [sequential, we end up with account numbers that are going to be random so there should not be any perceivable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=217) [pattern within those IDs that an attacker could then manipulate.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=222) [And again even if they did, so long as we maintain it on a per session basis, there's really not much damage](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=226)[they can do. Let's jump back to Visual Studio and implement an indirect object reference map.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=6&mode=live&start=232)

[Demo: Building an indirect reference map](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live)

[Now what I'm going to do this time is create an entirely new class to manage this.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=0) [So let's jump into the Solution Explorer and we will Add a New Item and that new item will be a code item of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=5) [type Class and let's call this IndirectRefMap and this IndirectRefMap is what's going to do all the work for us.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=15)[Now what I'm going to want to do is I'm going to use some extension methods here so it makes it really, really](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=25) [easy to add to our existing code and, of course, the first thing we're going to need to do then is make this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=30) [class static. Now there are three methods that we're going to create here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=36) [The first method that we're going to create, and it's going to be a static string, is GetIndirectRef and it's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=41) [going to take a directRef. So, in this case, this is going to be where the server says I have a direct reference,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=51) [so I have an ID from my internal database, and I would like to translate this into an external ID that I can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=62) [expose to my end user. So that's the first method we're going to need and we'll fill that in in just a moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=69) [The next method we're going to need will go the other way, so that's going to be, again, a public static string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=75) [So we're going to assume here that all of our identifiers are string, obviously the internal ID in this case](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=82) [is a userName, and we're going to create a cryptographically random string, which will be our external ID,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=87) [and this one is going to be GetDirectRef and it has the same method signature and we're prefixing it with this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=93) [because it's going to be an extension method and that's IndirectRef and we'll fill in the gaps on that in a moment too.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=104) [Now the third method that we're going to need and we can keep this one private, this is where we're going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=113) [actually build up and manage the internal map. So let's make a private static string, AddDirectRef, and it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=117) [could be a directRef like this. Okay, great. Now I want to start implementing this last one first because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=129) [it's the last one that really does the work. Now because we're going to want to create a cryptographically](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=136) [random string, I am going to be using System.Security.Cryptography and I'm going to create a new instance of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=141) [the RNGCryptoServiceProvider and we're going to want to create a little buffer here, which will be a byte array,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=154) [and the byte array size that we're going to use here, we'll make it a 32 byte array to hold our cryptographically](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=165) [random string, and now we're just going to fill that out of our RNG provider.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=171) [So, we will do rng.GetBytes and fill that into buff, okay, and now we have a nice byte array of random characters.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=175) [So now let's create an IndirectRef and what we're going to do with that IndirectRef, we want to push those bytes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=185) [into the IndirectRef and because it's ultimately going to go into a query string, we want to try and keep it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=192) [nice and URL-friendly so we'll use the HttpServerUtility and we will call UrlTokenEncode and we'll pass our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=199) [buff into there. Okay, that good. So now we have a string that we can use.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=209) [The next thing we want to do is we want to create a map and what I'm going to do in this case, and this is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=214) [map that we're going to persist in the session, is I'm going to create a new dictionary and the dictionary type](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=218) [will just be string, string. So, what we're going to do is add key and value pairs for both the direct and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=224) [indirect references. So let's say Add an Item, which will be the direct reference with the value that is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=232) [indirect reference, and then we'll go back the other way as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=242) [So it's easy to either pull this dictionary out by direct reference or pull it out by indirect reference.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=245) [Okay, that's great. So now that we've got that map, the next thing we want to do is push that map into session.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=252) [So let's push that into the Current.Session and we'll just call that session variable "Map" and we'll assign it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=260) [to map and the very, very last thing is that we need to actually return that indirect reference.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=269) [So what that means is, is that when this method is invoked and a direct reference is passed, it will add](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=276) [Dictionary items with a key both as the direct reference and the indirect reference, which will be our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=282) [cryptographically random string. It's a very, very basic method, but it's a good little demo for these purposes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=286) [Now let's go out and fill in these two methods. So, the first one is pretty straightforward; we will declare](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=293) [a map here. What we'll do is we'll cast that map first of all as a Dictionary type of string, string and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=301) [that is going to come out of HttpContext.Current.Session and, of course, we have called that "Map".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=309) [So, that's going to pull the map out of the session and cast it to a Dictionary type.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=321) [Now, let's put in a little ternary operator here so that if our map is null, then let's call AddDirectObjectRef](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=327) [and in that case, we'll pass the directRef and that will call the method we just created.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=334) [Otherwise, what we'll do is we will jump into that map that we've already created and pulled out of session and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=339) [we will grab the directRef key, which of course would return us the indirect reference value.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=345) [So, that's that one complete. The last one we need to fill in here is the GetDirectRef method.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=351) [Now, of course, when we're getting a direct reference, we're getting it via the indirect reference so we can't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=356) [just jump in and assume that we already have that indirect reference there.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=363) [In the previous method back up at getting the indirect reference, we were always able to create it if it didn't exist.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=366) [So, in this case, what we're going to do is, again, declare the map.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=373) [I'm not going to cast it to anything just yet. We're going to get an HttpContext and, again, we will grab](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=377) [Current.Session and we will grab "Map" like so. Now, if that map is equal to null, there's really not much](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=383) [we can do so the only thing that we're going to be able to do there is throw some form of exception and say](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=394) ["No map found" because, quite simply, there's no way of turning that indirect reference back into a direct](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=400) [reference if the map's not there. So, for example, if the session has recycled or if the session has ended,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=405) [we're not going to be able to pull that out and that is one of the risks of using session state to store this data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=411) [Conversely, if it is found, well then we can go back and we can cast it again and what I'm going to do is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=417) [create a new Dictionary and, of course, that is a string, string Dictionary and we will cast that against our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=424) [map and we will access it by key direct reference and that should be it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=430) [So, that's a pretty basic implementation. Let's now go through and implement that in the existing application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=438) [So the first place we're going to need to use this is back on our Index view and the reason why it's back on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=444) [that Index view, this is where we were appending the userName of the current user before.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=452) [Now this time, what we're going to want to do is actually call into GetIndirectRef.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=459) [So we're going to want to take that directRef and convert it to an IndirectRef and because it is an extension method,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=465) [we can now just do that and we'll need to add a reference here, so we'll add that guy, but the other thing we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=471) [might do here is that this isn't really a userName anymore so let's just make it a bit more generic and we'll](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=478) [call it an id. Okay, so the next thing we're going to want to do is get the directRef, which will be what we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=483) [do in the MVC API Controller. So, jumping back to the UserProfile, obviously the first thing that will be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=490) [different is we're actually passing through an id. Now, of course, that's going to break a few things because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=496) [we do actually need the userName so let's just jump down after the statement because, of course, we do want to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=502) [make sure the user is authenticated first and we will declare a userName and this time the userName is going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=508) [be equal to the id and then we'll call that extension method to get the direct reference and that should give us](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=515) [the userName. So that makes it very simple; it's just a one-liner here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=522) [Let's rebuild this and see how it looks in Internet Explorer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=526)[Alrighty, here we are back in IE. Let's now refresh this page and let's just take a look at what's in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=530) [source code here because we now should see something quite different to just a simple userName.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=537) [So, if we scroll down to where that API call is and, of course, now we've got id and this is a very different](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=543) [id to what a userName is. So obviously this is not something that an attacker is going to be able to readily](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=550) [enumerate through, this is precisely what an indirect reference should look like.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=556) [Okay, that looks good. Let's make sure that it works -- and there we go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=561) [So, the behavior is still identical to an end user. The big difference is, is that if an attacker starts poking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=566) [around in the source code or tracing out HTTP requests via Fiddler, they really don't have an id that they can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=573) [manipulate. Even if they could, we've now implemented the access control as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=580) [So, again, we've applied this security-in-depth mitigation where we've got proper authorization and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=585) [on top of that we've got an indirect object reference so really we've significantly strengthened](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=592) [the implementation of this app.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=7&mode=live&start=597)

[Obfuscation via random surrogate keys](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live)

[The last thing we'll talk about in this module on insecure direct object references is using undiscoverable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=0) [surrogate keys to add further obfuscation. So when we talk about a surrogate key, we're talking about something](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=6) [not made up of a natural entity so it's not like a name. An integer would be a surrogate key, but of course](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=13) [an integer does adhere to a predictable pattern. Something like a global unique identifier, though, a GUID,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=20) [whilst it does adhere to a pattern, the space that a GUID can occupy is enormous; they are considered global unique.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=27) [Of course, they're not actually global unique, but they're very, very close to it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=35) [So, if you were to use GUIDs as record or IDs, it would certainly mitigate the risk of insecure direct object references.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=39)[However, this should be viewed as security through obscurity. The only reason it adds any value is because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=48) [it's just a little bit harder to find. It's not actually implementing anything of any significance security-wise.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=56) [There are, of course, downsides as well. So, GUIDs are often found upon from the database side because they](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=63) [do not perform as well as small data types like an integer. Of course, they also have a high storage requirement.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=69) [It takes more space to store a GUID than what it does to store an integer, but most importantly, it just doesn't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=77) [change the requirement for proper access controls. No matter how far you obfuscate your keys and that includes via](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=83) [indirect object references, you still need to have the correct authorization on the resource that is being accessed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=8&mode=live&start=90)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live)

[Looking back at this module, the single-most important thing to protect your application from insecure direct](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=0) [object references is access controls. Does sufficient authorization exist on the record which is being accessed,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=6) [not just to ensure that the user requesting it is authenticated, but that they are actually authorized to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=14) [access the data. We saw before how an authenticated user can still go through and access records that they're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=22) [not meant to simply because the correct access controls haven't been implemented.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=29) [Indirect references are a great way of putting a layer of obstruction between the identities that we expose](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=34) [externally and the identities that we store internally. Particularly when those reference maps are session-based](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=41) [and the same IDs can't be used by anybody else independently of the access controls, that is a great way](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=49) [of adding a layer of security. Again, though, they're never a substitute for access controls and it's arguable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=56) [from most applications whether they actually need indirect object references.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=63) [A banking application, perhaps yes; your average line of business application or average website, probably not,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=68) [it does add a layer of overhead and in the case of storing things in session, there is a performance impact as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=75) [And finally, the surrogate keys can add some benefit in terms of obfuscation, but it really is security through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=82) [obscurity and there are downsides to using keys, such as GUIDs, in place of integers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=90) [So ultimately, it all comes back to start with the access controls, that's 95% of the problem, and go from there.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m4-direct-obj-ref&clip=9&mode=live&start=96)

[Cross Site Request Forgery (CSRF)](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to Part 5 of the OWASP Top 10 Cross Site Request Forgery.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=0) [You'll often also see this referred to as CSRF or sometimes XSRF.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=6) [So you may see the terms used a little bit interchangeably.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=12) [In this module, we'll start out by taking a look at how OWASP views the risk and how they categorize it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=17) [We'll then move on to performing an attack against an application at risk of CSRF.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=22) [After that, we'll dig a little bit deeper into the risk and try and understand how CSRF manifests itself and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=29) [how it can be mitigated with anti-forgery tokens. We'll then go and implement that in a vulnerable ASP.NET MVC](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=35) [application and we'll also take a look at how web forms approaches CSRF mitigation because it's quite different to MVC.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=43)[And finally, we'll take a look at some CSRF fallacies. There are some views on it out there which don't always](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=50) [hold true, plus we'll take a look at some of the native browser defenses, which can help mitigate the risk on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=57) [top of anything that we implement at the server level.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=0&mode=live&start=64)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live)

[Let's take a look at the OWASP overview and risk rating, and the first thing we'll notice here is that the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=0) [Threat Agents can be anyone who can trick the user into submitting requests to your website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=6) [We're going to refer to this word "trick" a lot of times throughout this module and as you'll see from the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=12)[Threat Agent, it could be any other website or HTML feed the user accesses that could perform this trickery.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=18) [The Exploitability is pretty average and the way this manifests itself is an attacker being able to create](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=26) [what we'd call a forged HTTP request and, again, we're seeing this word "trick" here, so trick the victim into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=34) [submitting a form and there are numerous different ways that this can be done and this is also where we see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=39) [the first reference about the user being authenticated. So usually with cross site request forgery,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=45) [we're talking about an authenticated user being tricked into issuing a request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=51) [The Prevalence of CSRF is very widespread, but the Detectability is also very easy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=58) [Now here you'll see a reference to predictability and it's talking about the fact that when we know how a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=65) [vulnerable website expects requests to be made, an attacker can go and forge those requests because it knows](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=70) [what pattern to conform to. Now, of course, if the user is authenticated and they're tricked into issuing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=76) [a request, then most of the time the browser will send things like cookies, along with the request, which is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=82) [where the authenticated user risk comes in to play. Fortunately, detection is pretty easy and we'll take a look](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=89) [at how to do that in just a moment. The Impact is being classified as moderate and it will really depend on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=94)[the nature of the vulnerable application. When you consider that CSRF is about the ability for an attacker to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=100) [trick the user into issuing a request, then really the scope for damage is pretty broad.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=107) [It's very dependent on what damage can be done by submitting requests to the target site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=112) [And finally, the Business Impact again is all about the nature of the business and the nature of the data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=119) [It will very, very much depend on the individual site and obviously exploiting that site may cause considerable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=125) [damage, not just to customers, but to things like reputation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=1&mode=live&start=131)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live)

[So here I have an ASP.NET MVC web application running locally on my machine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=0) [Now, this application would like us to update our status and now in order to update our status, we need to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=6) [log in first so let's jump over to the Log in page, and I'll log in with my User name and with my Password.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=13) [And once we are logged in, we will see that it is asking for a status.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=24) [At the moment, I do not have any statuses. So let's put in a quick status and we'll just give it a Good morning,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=29) [Update status, and now we are going to see that new status appear further down the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=36) [Let's now take a look at what's actually happening with the HTTP request behind this process.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=41) [So let's hit F12 to go to the Developer Tools and I'm going to jump over to the Network tab and I'm going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=48) [start capturing. Let's now make another update and I'm doing great and we're going to submit that update and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=55) [we'll see some requests appear and what we're really interested in is the first request, which is a POST request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=65) [So this is where our data is being sent to the server. Let's just go to the detailed view on that POST request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=72) [and we'll jump over first of all to the request headers and what we'll notice in the request headers is that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=79) [a cookie has been sent with the request. So this is our .ASPXAUTH cookie, which we've looked at in earlier modules,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=86) [and this is what keeps the session and the authentication persisted across independent requests in this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=93) [stateless protocol that is HTTP. So every time the browser submits a request to this domain, the cookie is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=99) [attached so that the user can be identified and their authenticated state persisted.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=106) [Now the other thing we'll see is that in our Request body, we have the status that we submitted.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=112) [So, what makes up this request is the header and data such as this is automatically attached and the body and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=117) [we've had to manually construct that in the website by virtue of creating a text field with a name of status](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=126)[and then as a user, we typed in this value. So, that's really very basic HTTP concepts.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=133) [Now let's go and have a look at how an attacking site could exploit a CSRF risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=139) [Now to exploit this risk, we need to hit a website with malicious intent.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=147) [Now let's imagine that we have a URL, such as the one in front of us.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=152) [It is a totally different domain, it's on the evilsite domain, and the path here is Winner.html.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=156) [Now this could come to us by any means. It could come to us through social media, it could come to us by](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=163) [clicking on a link in Facebook. How it comes isn't really the issue here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=168) [Let's take a copy of that, we'll grab a New Tab, paste it into the address bar, and let's now go and open that site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=173) [And as you can see, we are about to win something and this is always a very attractive proposition so let's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=180) [click on ‘Win now!!!' and yay -- we have won. Now everything that we have seen in this example just here is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=185) [on the evilsite domain. Let's say OK to that. Now let's go back to the Status updater and we're going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=193) [reload this page and what we can see here is that our status has been updated.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=200) [This is the current date and time and there is now a new message here and, in fact, this message also has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=207) [a link through to the same site and the same page that we were just on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=213) [So, this is a CSRF attack. This is a totally independent third-party site that has been able to trick the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=217) [browser into updating the status on our legitimate site. So, to better understand how this works, let's go](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=224) [and use the Developer Tools on the malicious site and have a look at how that request happened.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=233) [So I'm going to press F12 again and back to the Network tab, start capturing, and let's try and win again.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=239) [Now let's see what happens when we click OK. And here we can see another POST request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=248) [Now this all starts to look very familiar. In fact, it all starts to look quite like the request that we saw](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=255) [on the legitimate side. Let's go and open that first POST request, here's our cookie automatically being sent,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=259) [it has automatically been sent to the URL http://csrf, which is our legitimate site, and if we look at the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=267) [Request body, we can see that we still have a field called Status and now we have a malicious payload.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=275) [So, the point about this is that the request is actually an identical structure to the legitimate request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=282) [In fact, the only thing that's different in any of this is that the referrer is slightly different because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=289) [it's come from the attacking site. Now this is really important because what we've been able to do in this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=294) [demo is just a very simple reconstruction of the legitimate request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=299) [So this hasn't involved any sort of malicious attempts, such as SQL injection; we haven't exploited any](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=305) [cross site scripting flaws or other vulnerabilities in order to make this POST; it is simply that the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=310) [attacking site has been able to reconstruct the request to the legitimate site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=316) [And ultimately that's all that CSRF is really about -- the attacker tricking the browser into issuing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=321) [a request to the target site, which conforms to the structure of a legitimate request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=327) [The only difference is that in this case it had a malicious payload.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=2&mode=live&start=333)

[Risk in practice: Compromised Brazilian modems](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live)

[Let's take a look at a precedent of where CSRF is actually being used in practice for malicious activity](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=1) [and there's a very good example here about DSL modems being hacked in Brazil.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=7) [Now in this attack, the first thing that victims observed was when they went to Google.com.br, which is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=12) [legitimate address for Google in Brazil, they were prompted to install some software.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=20) [Now in this case the software was referred to as Google Defense and they required it in order to access the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=25) [new Google. Now what was actually happening here was routers were being exploited and the DNS servers changed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=31)[So what it means is, is that once the DNS server is changed and pointed to a malicious address, a request for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=40) [a domain, such as Google.com.br, could actually be routed to another server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=47) [So you'd see the correct address in the browser, but the request would actually be sent to a malicious site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=53) [Now all of this was possible by exploiting a cross site request forgery vulnerability within the router itself.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=59) [So Brazilian users were going to a website, which would trick the browser into issuing a request to the router](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=67) [and that request to the router would change the DNS settings as we can see here in the screen grab.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=74) [So, in this example, it was actually a chaining of attacks. So firstly, there was a CSRF vulnerability and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=80) [secondly, there was the malware that was served from the site, which appeared to be a legitimate Google domain.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=88) [Now once the malware was installed, the attackers were able to commercialize the operation and that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=94) [commercialization could mean any number of things, such as installing key loggers to capture key presses,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=100)[capturing sensitive data that the user enters, spying upon passwords and banking information, any number of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=106) [different ways of exploiting a user once their PC is infected with malicious software.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=112) [So, again, this is a good example not just of how much damage a CSRF attack can do, but how attacks are combined](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=119) [together, CSRF plus malware, and many of the hacks that we see do involve the chaining together](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=126) [of multiple risks in order to exploit victims.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=3&mode=live&start=133)

[What makes a CSRF attack possible](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live)

[Now that we've seen a CSRF attack in practice, let's have a look at what makes it possible in the first place.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=0) [As we just saw in the demo, the main thing about the CSRF attack is the malicious request was structured just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=7) [the same as a legitimate request. So included in that malicious request was an authenticated session,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=15) [which was being persisted via the ASPXAUTH cookie. Now, of course, this cookie is going to be sent with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=22) [every request to the domain that it was set for. That's the way cookies work.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=29) [Now because of that, the attacking site was able to recreate the legitimate request, except this time it was](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=34) [able to do so with a malicious payload. And obviously in the case we just looked at, that malicious payload](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=40) [was formed data. So once the attacking site was able to construct a request that conformed to a legitimate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=47) [pattern, albeit with a malicious payload, all it took was for the victim to visit that attacking site and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=53) [the request was automatically issued. So let's go and take a look at the code on both our target victim site](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=59) [and also on the attacking site and let's see how this actually happens at the code level.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=66) [Let's now take a look at the application we exploited a little bit earlier on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=73) [So this is the victim application, the one where the cross site request forgery risk was exploited, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=77) [the first thing to note here is this is an ASP.NET MVC 4 application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=84) [Now the same risk exists in ASP.NET web forms, but the way we mitigate it is actually a bit different and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=90) [we'll look at how we do that a little bit later on in the module.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=96) [So the first thing I want to point out with this application is that it is an ASP.NET MVC controller,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=100) [this is the HomeController, and what I want to look at is that we have a Controller action called Index,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=106) [and this particular Controller action is decorated with an HttpPost data annotation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=114) [So in other words, this is the Controller action that is going to be called when we post to the root of the site](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=120) [and that's the path that we saw attacked earlier on in the Internet Explorer Developer Tools.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=126) [Now what we'll see with this Controller action is that it takes a model of type StatusUpdateViewModel and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=132) [I just want to drop into this model and take a quick look at what it does inside.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=138) [So we will go to Declaration and the thing to note with this model is that it takes a string named Status.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=142) [So in other words, when we post to that Controller action, so again that is just the root of the site,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=149) [we need to post a StatusUpdateViewModel or in other words, we need to make a post with a piece of form data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=156) [named Status, and then that will just automatically map through to this attribute on the StatusUpdateViewModel.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=163) [Now let's just jump back into the HomeController and the next thing I wanted to show you on this particular](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=169) [Controller action is that there is an Authorize annotation. Now this is important because what it means is,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=175) [is that this Controller action can only be called by an authorized user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=182) [Now this is important because what it means is, is that the identity of the user calling this Controller action](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=187) [will be verified when it is invoked and at several times throughout this module, we've seen reference to how](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=193) [CSRF tricks authenticated users into making unintentional requests, so that authenticated state is quite](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=199) [important in this CSRF example. So let's take a look at what's actually happening in this Controller action](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=208) [and the first thing that happens is the model is validated. So that's a pretty typical thing to do at the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=214) [start of a Controller action, nothing new there. We are then getting the userId of the currently logged-in user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=220) [So this comes back to that Authorize annotation. We're going to make sure that the user is authenticated and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=227) [then we're going to grab their username. Now the username is important because that's the person we're going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=232) [to want to save this StatusUpdate against. Now what we'll see on the next line is that we're calling into our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=238) [Entity framework data context, which we've named StatusEntities, a StatusUpdate is being added, and that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=244) [StatusUpdate is constructed from the ViewModel and this is where we're passing through that Status.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=250) [So going back to our model, there is an attribute on the model called Status.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=256) [All it needs to set that value is a post request with the piece of form data called Status and then it's also](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=259) [setting the userId, so again here's this authorization component where it's saying, hey you are actually logged on,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=266) [we are going to do something under your identity, and save some data against your identity, and it's just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=272) [defaulting the time to the current time. That gets saved in the data context and then there's a redirect back](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=278) [to the index. So that's going to cause an HTTP 302, which means the browser will just make a GET request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=284) [back to the homepage and, in fact, that will then go back up and just load the Index controller action.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=290) [So the point of all this is that this really, really is a basic feature.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=296) [It looks like a very normal implementation of code and there's nothing in there that looks particularly vulnerable.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=300) [Likewise, there's nothing in there that really looks like a CSRF defense and that's something we'll come back to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=307) [just a little bit later. Let's now go over and look at the attacking site and see how it was able to take](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=312) [advantage of this implementation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=318)[Okay, so this is the page, which was attacking our site in the earlier CSRF example, so this is the Winner.html page,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=323) [and there's a few things I want to point out here. Now the first thing is, is that we have a form and the form](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=331) [has an action, which is the path of our legitimate site. So clearly this path will load the default action on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=337) [the default controller, which is the one that we were just looking at.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=344) [And, of course, because the method is post, it's going to load the one that was decorated with the HTTP post annotations.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=347) [So in other words, the one we just looked at is going to save the StatusUpdate.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=354) [Now when this form posts, it is going to target a frame called csrfFrame and we'll see that down here a little](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=357) [bit further that we have an iframe called csrfFrame and it is hidden.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=365) [So the visibility has been set to hide the contents of that frame. So what it means is, is that when this form](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=370) [posts, whatever the response is from the target site isn't going to be visible to the user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=377) [Looking inside this form, we have two input controls. The first input control is hidden and it has a name of Status.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=383) [So this Status aligns to the Status attribute on the ViewModel, which we looked at earlier on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=391) [So normally you would have an HTTP post from the legitimate website, it would be sending a piece of form data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=398) [named Status, and it would line up with what the controller action expects from the model.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=404) [And all that's happening here is that this page is reconstructing that pattern.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=409) [Now, of course, what it's doing is it's doing it maliciously so this is why when we look at the value,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=414) [we can see that this is the malicious string that we saw earlier on, so this was the StatusUpdate which we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=419) [never intended to make. This StatusUpdate also passes through some HTML, it passes through a hyperlink tag,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=425) [and we saw earlier on that this actually rendered into the source code and we ended up with a hyperlink saying,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=432) [click here, as opposed to the whole thing rendering to the screen.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=437) [So clearly there's also a cross site scripting risk because of lack of output encoding.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=441) [Now that's another risk, but it does show how attackers can chain risks together and in a case like this,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=446) [actually use the cross site scripting vulnerability to self-propagate the attack.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=452) [So obviously if somebody else sees this link and clicks on it, they're going to end up with the same StatusUpdate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=457) [and the whole thing keeps going round and round and round and we often see attacks like this through mediums,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=463) [such as social media, where the likelihood of it propagating is very, very high due to the large number of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=468) [connections that people make with other people. So, moving on, there is a Submit button, which is our second](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=474) [input field and clearly that is the one that said "win now!!!" earlier on and what we'll see is that when we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=480) [click it, there's going to be a JavaScript alert, and it says "Yay, we've won!!!"](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=485) [Now, of course, what it actually means is that the alert will fire first, we'll see that box, you click on OK,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=490) [then the form will submit, it'll post the malicious StatusUpdate to the vulnerable site, and the user is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=496) [none the wiser because the response goes into that hidden frame.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=502) [So that's the entire execution of this attack and it's made possible simply because all this site does is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=506) [reproduces a legitimate request and tricks the user's browser into issuing it whilst they are authenticated](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=512) [and that is precisely what a CSRF attack is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=4&mode=live&start=520)

[Understanding anti-forgery tokens](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live)

[The single most predominant way of protecting against the risk of CSRF is to use what is referred to as an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=0) [anti-forgery token. So let's take a little bit of look at what an anti-forgery token is and then we'll move on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=7.5) [to implementing one. Now, first of all, we need to remember that CSRF attacks work because they're predictable.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=14.5) [Now when I say predictable, I mean that a CSRF attack simply replays a legitimate request, albeit with a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=22.5) [malicious payload, and we saw this a little bit earlier on. So the request was issued with legitimate cookies](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=29.5) [because the cookies were correct for the domain and it was obviously issued to a legitimate domain and a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=36.5) [legitimate path; the only difference was that the malicious request embedded a custom payload, which is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=41.5)[obviously set by the attacker. Now what a CSRF token or an anti-forgery token does is it adds randomness to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=48.5) [the process. When we talk about a token, what we're really talking about is a random string and in the case](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=56.5) [of an anti-CSRF token, we're talking about a piece of data known by both the legitimate page where the form is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=64.5) [based and a piece of data that can be sent via a cookie in the request header.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=71.5) [Let's take a look at how that workflow looks.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=76.5) [So to better understand how an anti-forgery token actually works, let's imagine a request and a response sequence.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=81.5) [So we begin with the user requesting a page from a server; nothing unusual about this, this is how our example](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=89.5) [earlier on worked. Now what's a little bit different this time is that when the server responds, it has a token](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=95.5) [in a hidden field in the form of the page, plus it also has a token in the response header via a cookie, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=103.5) [we'll look at this in a moment, but when that response comes back to the server, we're actually going to get](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=111.5) [two tokens. Now when the form is submitted, the token in the hidden form field is sent, but also the token](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=115.5) [in the cookie is sent. Now this is important because we know that any cookie set against the domain is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=124.5) [automatically going to be sent, but what's different to earlier on is that hidden form field and this is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=131.5) [randomness because the attacking site doesn't know what value the token should be in the hidden form field.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=137.5) [Now once both of these tokens get submitted to the server, the server can then go on and validate them,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=144.5) [make sure that they match, and if they don't then it can reject the request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=150.5) [So this is a very, very effective way of mitigating against the risk of a CSRF attack, in fact, it's really](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=155.5) [the only solid defense that we have against CSRF. Let's go back to our ASP.NET MVC application and actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=160.5) [implement an anti-forgery token so we can see precisely how it works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=5&mode=live&start=169.5)

[Demo: Implementing an anti-forgery token in MVC](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live)

[Let's now implement an anti-forgery token in the vulnerable application, which we exploited a little bit earlier on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=0) [Now with ASP.NET MVC, it's extremely easy, and we need to actually implement this in two different points.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=7) [So the first thing we need to do is in our view, we're going to add an HTML helper and this HTML helper is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=15.5) [called the AntiForgeryToken and all we do is drop that within the using statement for our Html.BeginForm method.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=23) [So that's all we need to do here and what it's going to do is, this is going to create a hidden form field](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=32) [with a token in it. It's also going to ensure that when the view is loaded, the token is set in a cookie](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=39) [in the browser so that it comes back when the form is submitted.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=45) [So what it means is, is that we'll get a token back via a form post body and we'll also get a token back via a cookie.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=48) [Let's jump back over to the HomeController and what we're going to do now is we are going to add another annotation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=56) [and this annotation is going to be ValidateAntiForgeryToken.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=63) [Now what that means is, is that when this Controller action is called, ASP.NET MVC is going to make sure that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=69) [we've received a valid token. If it only receives a token via the cookie and it doesn't receive that hidden](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=77) [form value token or it doesn't receive any token at all, an exception is going to be raised and we'll see what](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=83)[that looks like in just a moment. So let's build the application and then we will jump back over to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=88) [the browser and see what it looks like.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=93) [Okay, back in the browser and I am still logged in. Let's now refresh and we will go through and enter a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=98) [new Status and this time I am going to do a ‘Good evening', Update status, and we now have ‘Good evening' and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=104) [that has loaded just fine. So obviously legitimate use still works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=115) [So like we did before, let's go and issue this legitimate request again and this time we will hit F12 to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=120) [load up the Developer Tools and we'll start capturing because I want to take a look at what's actually happening](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=127) [in the background. So, ‘I'm still great', let's update the status, and here we see a series of requests.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=132) [And what I want to look at this time is, once again, the first request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=140) [Now the first thing you'll notice here in the Request body tab is that we now have a RequestVerificationToken](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=145) [and you'll see we have a great big ID in here, which is the majority of the Request body and, of course,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=151) [we still have our Status field. So this is what we had before.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=157) [Obviously the difference here, the major difference, is that this Request body, in this post request, has this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=161)[great big RequestVerificationToken. Now this is really important.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=167) [Let's now take a look at the cookies and what we'll see now in our cookies is that we have a RequestVerificationToken.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=172) [So we've had two tokens sent by two different channels -- one in the Request body, one in the Cookie.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=179) [Now let's go over to our attacking application and try this again and we'll see a very different result](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=186) [to what we had before.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=192) [Okay, so here we are back on the familiar evilsite and it is identical to before, nothing has changed on the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=196) [evilsite since we implemented the AntiForgeryToken. Let's now see how this thing works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=202) [So, we're going to do a ‘Win now!!!' and we get the same Alert box, OK to that, the end user experience is identical.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=208) [Let's now go back to our target site, we'll give it a refresh, and we don't have any malicious StatusUpdate;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=217) [it has not worked. So, remember, these are the ones that we had earlier today.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=227) [Now when we try and run it, it is not there. So, to understand this a little bit better,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=231) [let's jump back to the malicious site, let's hit F12 for the Developer Tools again, Network, Start capturing,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=236) [let's try and ‘Win now!!!' again, OK, and now we can see that there is actually an HTTP 500 response when](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=246) [the malicious site attempts to post to the target site. Let's open up this request and let's actually have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=256) [a look at the Response body this time. Now, of course, we've got a lot of HTML here, but here is the important bit.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=263) [So in the Response body, the website has responded, "The required anti-forgery form filled", which is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=271) [RequestVerificationToken, is not present and, in fact, you can see that the server has responded with an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=278)[HttpAntiForgeryException. So this is very, very important. If we have a look at cookies, we can see that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=285) [still have the RequestVerificationToken because, of course, cookies get automatically sent back to the domain](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=293) [that it was set for, just like the ASPXAUTH cookie; the big difference is, if we look at the Request body,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=299) [we only have the Status -- we don't have that AntiForgeryToken because there's no way that the attacking site](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=305) [could actually know what it is. So this has put a complete halt to the ability for that malicious site to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=312) [actually take advantage of the CSRF risk and the best thing about it, in ASP.NET MVC, is it was only two lines](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=319) [of code; it was just the single line in the view to add the HTML helper for the AntiForgeryToken and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=327) [there was a single annotation on the Controller action, which validated the token. That's very, very simple.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=332) [Let's go and take a look at how it works in ASP.NET web forms.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=6&mode=live&start=339)

[Demo: Web forms approach to anti-forgery tokens](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live)

[I've just created a brand-new ASP.NET web forms application, it's an ASP.NET 4.5 application obviously created](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=0) [in Visual Studio 2012, and what you're seeing here is some default CSRF defense, which is implemented by the template.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=8) [Now what we're looking at is the code behind of the master page and what you're seeing is a very manual](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=17) [implementation of the same sort of defense that we just saw in ASP.NET MVC.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=24) [So let's take a quick look at how the implementation works when it's very manually constructed and the first thing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=29) [you'll see here is that the Page\_Init event goes away and actually looks for a cookie called AntiXsrf](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=35) [(so there's that alternative acronym to CSRF) TokenKey, and of course that's just a string value like so.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=43) [Now what it tries to do is check that first of all it does exist and second of all that it can be parsed with GUID.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=50) [So what's happening in ASP.NET web forms is they're using GUIDs as a random string to generate the token.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=56) [If it exists, it stores it locally and it also puts it in the ViewStateUserKey of the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=63) [If it doesn't exist, then it simply creates a new one and a new one is just a new GUID and it puts that in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=69) [the ViewStateUserKey. It then goes through and creates a Cookie with the token, it sets it to HTTP-only so](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=75) [that it can't be accessed by client-side script, sets the value; if we're using SSL and the connection is secure,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=82) [it also makes sure that the cookie is secure so that the cookie won't get sent back over an insecure connection,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=89) [and it gets set in the response. And then finally there's also another event added to the Page.PreLoad event,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=95) [which calls master\_Page\_PreLoad, and that event exists just a little bit further down the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=102) [So this is the really important bit because this is where ASP.NET web forms is actually validating the token.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=110) [Now if it's not a PostBack, then in this case it's not worried about CSRF, so it simply sets the token back](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=116) [in the ViewState with the TokenKey and also the XsrfUserName and the username in this case is the identity](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=122) [of the current logged-on user. Now one thing to remember is that you can have a CSRF risk for GET requests](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=129) [as well, so it doesn't necessarily have to be a PostBack, but the implementation of an AntiForgeryToken in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=137) [a GET request is a bit more of a bespoke requirement that we can't really create a generic model like this for.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=144) [Now, alternatively, if it is a PostBack, then ASP.NET web forms is going to make sure that the token passed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=150) [view in the ViewState, because remember we stored the token in the ViewState earlier on, matches the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=157)[antiXsrfTokenValue. Now if it doesn't or if the AntiXsrfUserNameKey, so the key specific to the current user,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=161) [doesn't match the current identity, then we have a problem. So that's when our InvalidOperationException is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=171) [going to be thrown and this is effectively where our request validation token is validated and it doesn't add up.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=177) [So this is when web forms will throw an exception and obviously not process the request any further.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=185) [Now this is obviously a rather manual construction of an Anti-XSRF token, it certainly is very manual compared](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=190) [to what we saw in ASP.NET MVC just a little bit earlier. Fortunately, this is implemented in the default template](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=198) [though from a new web forms application, so it's manual in terms of the underlying code is still clearly visible](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=205) [and compiled as part of the application; on the other hand, at least it's already created and a brand-new](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=211) [ASP.NET web forms application will be able to take advantage of Anti-XSRF defenses.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=7&mode=live&start=216)

[CSRF fallacies and browser defences](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live)

[CSRF tends to be one of those attack vectors, which can be a little bit misunderstood, and certainly there are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=0) [some fallacies out there about how you should best protect your application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=6) [One of the recommendations is to implement referrer checking to ensure that a third-party site can't actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=11) [post to the target site. Now there is some legitimacy to this particular approach and certainly earlier on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=18) [when we had the Developer Tools open and we looked at how the attacking site made the request, we actually saw](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=24) [that the referrer was not the legitimate site we're posting to, obviously it was the attacking site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=29) [So, you could actually implement referrer checking. The problem, of course, is that you then need to implement](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=36) [referrer checking in every location where you're going to actually post data and the other thing is,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=41) [it doesn't mitigate the risk of a CSRF attack coming through a vector, such as cross site scripting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=46) [So, for example, if there's an XSS risk on a website, there's no reason why an attacker couldn't embed a CSRF](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=53) [attack sequence into that site and if that XSS is then either reflected or persisted, certainly the referrer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=60) [will actually be a legitimate site. So effectively it is attacking itself via XSS and then referrer checking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=67) [won't really help. Another view is that you should disable the HTTP GET verb on at-risk pages, so one CSRF attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=73) [that we often see is a vulnerable URL may be used as say the source of an image.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=82) [So when the browser loads the image, it naturally makes a request for the image source and if that image source](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=88) [is a path which is at risk of a CSRF attack via a GET request, then the page could be exploited.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=93) [So certainly you could disable HTTP GET on pages which only require POST, but as we just saw, it's extremely](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=100) [simple to create a CSRF attack using a POST request. Another misconception about CSRF is that validating the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=107) [IP address of the browser loading the form matches the IP address of the browser posting the form offers some](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=115) [sort of protection. Of course, it doesn't really offer any protection because in a CSRF attack, the victim's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=123)[browser is the one both loading and posting the form and, indeed, this is one of the things that makes it so effective.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=128) [For all intents and purposes, it looks like a legitimate sequence of requesting a form and then posting it back.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=134) [So none of these things really make a fundamental difference. Some of them might help a little bit, but ultimately](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=140) [the only real defense against CSRF is using anti-forgery tokens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=145) [Another thing worth touching on briefly is native browser defenses.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=153) [So, earlier on, we looked at how Internet Explorer had some quite good cross site scripting defense,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=157) [which helped prohibit things like script tags being injected into HTML.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=162) [So in the case of CSRF, often we will see defenses around things like Cross-Origin Resource Sharing or CORS,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=167) [which is intended to prohibit one domain maliciously interacting with another domain.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=174) [Now there is some benefit to that, but of course, as with most things in security, there are also work-arounds](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=179) [and other attack vectors. Most importantly though, and just as we discussed back in the cross site scripting module,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=185) [you should never rely on the native browser defenses to save someone from a CSRF risk on a web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=191) [It always comes back to making sure that you've implemented robust security measures on the individual website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=200) [So always code as though no browser defenses whatsoever exist.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=8&mode=live&start=207)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live)

[Let's summarize the module.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=0) [The most important thing we discussed in terms of understanding how CSRF manifests itself is that it's all about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=2) [an attacking site tricking the browser into issuing a request to the target site, which looks entirely legitimate.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=9) [Of course, the difference is, is that attacking site is issuing a payload with malicious intent; it's tricking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=17) [that user's browser into issuing a request, which the user had no intention of actually making.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=23)[Now because our authenticated sessions tend to be persisted via cookies, the attack also exploits the fact that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=29) [cookies for a domain will be issued in every request the browser makes for a resource on that domain.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=35) [So it's particularly malicious in that it's taking advantage of the authenticated state of the user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=42) [Now as we saw, the only real defense for a CSRF risk is by using anti-forgery tokens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=48) [When we looked at it in MVC, it was very easy. There's an HTML helper and then there's a validation annotation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=54) [that can be put on the Controller action. In web forms, it is messier.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=60) [We don't have those same sort of native controls and native defense built into the framework; however,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=64) [at least a new ASP.NET web forms application provisioned from Visual Studio 2012 does have some defenses](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=71) [built into the code behind of that master page. So although it's a little bit messy, at least we do have a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=78) [defense in there, but anti-forgery tokens are really about the limit.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=84) [Because a malicious CSRF request mimics a legitimate request, it's very difficult to build other defenses](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=88) [around the risk of cross site request forgery. And finally, like we saw with cross site scripting, browsers do](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=95) [offer some native defenses and it's great that the browser vendors are building in this extra layer of defense](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=103) [for consumers, but just like we saw with cross site scripting, we should never rely on the native defenses](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=109) [in the browser. Those defenses do differ from one browser vendor to another, it's better in some, it's worse](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=115) [in others, but they do also change and exploits are found that can circumvent some of the controls in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=121) [So always code like those defenses don't exist and focus on securing your web application and really for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=129) [cross site request forgery, that means using anti-forgery tokens to validate that all the requests that cause](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=136) [changes on the server, such as posting data, are actually legitimate.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m5-csrf&clip=9&mode=live&start=141)

[Security Misconfiguration](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to Part 6 of the OWASP Top 10 on Security Misconfiguration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=0) [In this module, we're going to start out by taking a look at how OWASP views the risk of security misconfiguration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=8) [We will then move on to performing an attack on a vulnerable application that has a couple of problems with it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=15) [After that, we'll look at correctly configuring custom errors and tracing in the application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=21) [We'll also take a look at how we can keep packages current with NuGet and keeping packages current is an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=27) [important aspect of security misconfiguration. We'll then take a look at what's involved in encrypting sensitive](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=33) [parts of the web.config so that they can't be read outside the server environment, which they deploy to.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=40) [And we'll take a look at using config transforms to try and avoid the risk of a security misconfiguration flaw](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=46) [slipping through into a deployment. And finally, we'll also check out what it means to enable retail mode](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=53) [on the server. So all of these are what OWASP would refer to as various aspects of security misconfiguration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=0&mode=live&start=60)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live)

[Let's take a look at the OWASP overview and risk rating. As usual, the Threat Agents are pretty broad so](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=0) [consider any external attackers, as well as people who may actually have accounts in the system.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=7) [All of these are considered possible threat actors for security misconfiguration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=13) [When we look at the Attack Vectors, we can see the exploitability is easy and what you'll see here is a broad](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=19) [range of possible Attack Vectors, so things like default accounts, unused pages, unpatched floors, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=25) [unprotected files; there are many different ways that the risk of security misconfiguration can be exploited.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=32) [That actually makes this one of those more generic risks in the Top 10.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=40) [So there's not one single thing that we can point at as security misconfiguration, rather a broad range of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=44) [individual discrete risks. When we look at Security Weaknesses, we can see that the prevalence is common,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=50)[but the detectability is also easy and the other thing you'll see here is that security misconfiguration can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=58) [happen at various points across the application stack. So, it could be at the web server or at the application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=64) [server or framework or at some point in custom code. There are lots of different points where the risk of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=71) [security misconfiguration may be present. Now because the detectability is quite easy, it also means that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=78) [we can use things like automated tools to detect common security misconfiguration risks, so that does help us](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=85) [a little bit in the mitigation. OWASP classifies the Technical Impact as moderate and inevitably the impact](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=91) [is always going to be dependent on the type of data that is being protected, but certainly there are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=99) [security misconfiguration risks, which could actually lead to the complete compromise of a system.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=105) [So it can be a very serious flaw as well. The Business Impact is, of course, always dependent on the nature](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=110) [of the business, but the risk of security misconfiguration can lead to attackers being present in a system](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=117) [for quite some period of time before it's actually detected.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=1&mode=live&start=123)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live)

[Let's now take a look at the risk of security misconfiguration in an ASP.NET web forms applications.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=0) [So I've got a demo app here; it's a very basic app built on the standard Visual Studio 2012 template for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=6.5) [ASP.NET web forms and the only thing that I've really added here is I've added a little section here on Widgets.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=12.5) [And when we go into Widgets, we will see a list of widgets. Now, in this case, our widgets appear to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=19.5) [Mountain Bikes and what I want to do with this application is start to probe a little bit and see if I can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=25.5) [discover some more information about it and this is typically what an attacker will do -- they'll try to disclose](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=30.5) [internal implementation information so that they can probe for weaknesses.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=36.5) [So let's start by having a look at this query string and we can see here that I have a query string, which is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=41.5) [an integer. So one thing an attacker might do is see what happens if that query string is no longer an integer.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=46.5) [So let's change that to an "x" and we might load that page and here's the first piece of security misconfiguration](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=53.5) [that we can see. Now in this case, what's happening is that the application is disclosing internal information.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=61.5) [So, we can see the error message that the framework has omitted, so obviously our input string is not in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=68.5) [correct format; we can see the internal code complete with line numbers and the actual syntax of the code;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=74.5) [we can see where the source code is actually stored and in this case, I've got it sitting in my Dropbox Pluralsight;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=80.5) [then we get some internal Stack Trace information -- FormatExceptions, input strings, and then of course we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=87.5) [also get the .NET framework version and the version of ASP.NET. So there is a lot of information that is being](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=93.5) [disclosed by this error message. Now one thing an attacker wants to do is to try and probe with data,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=100.5) [which is probably outside accepted ranges, and see how the application responds.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=108.5) [So obviously we've managed to call an internal exception and we've been able to leak lines 13 through to 17](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=114.5) [just here by passing a string rather than an integer. One thing that's interesting about this code is that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=119.5) [we seem to be parsing an ID as a 64-bit integer, which is a little bit unusual.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=125.5) [An attacker might now start to say, well Int64 is quite an odd data type for an ID, maybe this should really be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=132.5) [a 32-bit integer. What would happen if we had an integer which was somewhere between 32-bit and 64-bit?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=138.5) [So let's maybe take a look at that. Now we know a 32-bit integer can hold a number of just over 2 billion,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=145.5) [so what if we started to manipulate this input parameter and we made it say 3 billion something.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=152.5) [So 3, 1, 2, 3 - 0s, another 3, and another 3, and let's see what happens now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=157.5) [And here we see that another exception has been raised and this time we have got an exception about the value](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=166.5) [being either too large or too small and we know it's actually too large for an Int32.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=172.5)[So what we can see here is that this untrusted user data is actually being cast to a couple of different types.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=178.5) [It was okay for an Int64; it's not okay for an Int32. So what it's now doing is disclosing some more information](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=185.5) [about the application and, in fact, what it's disclosing is that we can see that we're working with some](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=192.5) [variable called Command and when we look at the fact that it's adding parameters and it's opening a connection,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=198.5) [it's probably pretty safe to assume that this is a SQL command and obviously, here, we can see that the data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=203.5) [type is SqlDbType.Int. So, it's given us a little bit more information and we can see that this TypeId is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=209.5) [actually being used and if we go back to the previous page, we can also see that that TypeId is being requested](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=215.5) [in the QueryString and we can also see that it is being converted to an Int64 just the same way as our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=221.5) [CategoryId was. So we're not actually seeing TypeId passed in the QueryString for the page that we just hit,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=228.5) [but clearly it is being read in by the code. So, as an attacker, let's go and add a TypeId and we'll add an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=235.5) [ID which won't cast to an Int64 and let's see if we get a little bit more information out from a new exception.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=244.5) [So let's go back and change this to a valid CategoryId again, but this time we will manually just add in a TypeId](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=251.5) [and let's maybe make it another "x". So we'll make it a string, which won't convert to an integer, which should](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=259.5) [cause another exception, which should bubble up to the browser, and here we go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=265.5) [So now we've got another exception and we've got a couple of other lines of code in here that we didn't have before.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=271.5) [So, for example, on Line 19 we can now see what the SQL statement was and one thing we can see about this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=278.5) [SQL statement is it's actually parameterized. So in this case, the developer has successfully mitigated the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=284.5) [risk of SQL injection, which is a good thing; however, what we can see on Line 18 is we can see a connString.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=290.5) [Now this is a serious risk and, in fact, if we have a look just a little bit further across, we can see that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=298.5) [we've got the Data Source, we've got the Catalog -- so the database name, the userName, and most importantly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=305.5) [we've also got the Password. So due to security misconfiguration, this application is leaking enough data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=312.5) [for an attacker to remotely connect to that database. Now, of course, that's assuming the database allows](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=321.5) [remote connections and doesn't just accept connections from the local app, but in many cases this would be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=326.5) [an exploitable risk. It is very, very common for databases to allow those external remote connections.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=334.5) [Now, of course, even if it didn't, it's never good to leak this information, but what it means is, is there's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=341.5) [a high probability we could now jump over into a tool, like SQL Server Management Studio, and connect with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=346.5) [this exact connString. So this is a really serious case of security misconfiguration, which could lead to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=352.5) [some serious damage to the application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=2&mode=live&start=359.5)

[Risk in practice: ELMAH](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live)

[Let's take a look at an industry precedent of security misconfiguration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=0) [Now this time I would actually like to look at a risk, which is present in many ASP.NET websites today, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=5) [that risk is the use of an unsecured ELMAH handler. Now ELMAH is a very popular third-party library for ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=12) [It's the Error Logging Modules and Handler Library and what ELMAH does is it automatically collates unhandled](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=22) [exceptions that happen in an ASP.NET application, stores them, and makes them available for review by a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=29) [system administrator at a later date. Now unfortunately what happens with ELMAH is that many times it is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=36) [incorrectly configured and what it means is, if we do a Google search for ELMAH and, in fact, we can do that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=44) [by looking for the presence of the elmah.axd handler in the URL and we also search for the presence of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=50) [string text "error log for", which appears on the index of every ELMAH log handler.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=59) [What we find is that there a huge number of results, which are publicly accessible.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=65) [Now ELMAH attracts a huge amount of very valuable information; in fact, it's a very useful library for developers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=71) [It's also a very useful library for attackers. So, for example, if we were to change this search term](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=79) [a little bit and let's include aspxauth and, if you recall, aspxauth is the name of the cookie which actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=85) [tracks the authenticated state of a user in an application using ASP.NET forms authentication.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=94) [Now in an earlier module, we also saw that if we could get the value of that cookie, we could hijack the session](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=100) [and, in fact, what we'll find is that there are many, many different sites that do have the ELMAH log exposed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=106) [and have the presence of the ASPXAUTH cookie. So, for example, just picking the first one at random, if we have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=113) [a look at the ELMAH log of this site, we will find that the ASPXAUTH cookie value is accessible by anyone.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=119) [Now fortunately, this one is actually quite old, but if the ASPXAUTH cookie was still active and the session](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=128) [had not expired, all it would take is for an attacker to literally copy this and paste it into a new cookie.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=134) [This would give the attacker the exact same rights that the original user currently has.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=140) [Now even if they couldn't find one for a recent website, it would only take a little bit of social engineering](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=146) [to get an authenticated user to cause a server exception and then the attacker could come and grab this value.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=151) [So there's a very, very serious risk here. There's also a serious risk in that there are many instances where](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=157) [ELMAH is leaking other sensitive data. So, for example, it's leaking the connectionString like we saw in our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=163) [own example a little bit earlier on. There are also many examples where it's leaking passwords because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=169)[developers are actually capturing passwords and exceptions and then those become visible in the ELMAH log.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=174) [So, ELMAH is a fantastic tool, but unfortunately there is a huge prevalence of security misconfiguration and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=180) [it's a very good example of just how bad this risk can be.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=3&mode=live&start=187)

[Demo: Correctly configuring custom errors](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live)

[Let's now take a look at what it is that made this web application so vulnerable and what it really boils down](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=0) [to is the fact that when an internal exception occurred, that exception was exposed publicly.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=6) [Now, of course, a developer should do their best not to have any internal exceptions and do things like make](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=13) [sure that types are valid before trying to cast them, but you need to be able to handle it when an exception](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=18) [does occur and by handle it, I mean not disclose internal implementation data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=24) [So what I want to look at now is how to configure custom errors correctly.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=29) [So, I'm looking at the web.config here of the application and what I want to do is inside system.web is add](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=35) [a customErrors node. Now, what we'll find with customErrors is that there are a few different configurations](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=41) [we can use to secure the application a bit further and the first thing we want to do is turn customErrors on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=47)[So we've actually got three different options -- there's Off, which is the default and that's what allowed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=55) [the internal error to bubble up to the browser earlier on; On, which we'll see in a moment; and RemoteOnly,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=60) [which turns it on, but only when it is running on a remote server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=67) [So, if you're running it locally in your development environment, it would be off; if you publish it to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=71) [the server, it would be on; so RemoteOnly is a pretty good sort of safety net.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=76) [For the moment, we're going to turn customErrors On, we're going to save the web.config, and let's see how](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=80) [this changes the behavior of our application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=87) [Okay, so let's now go back to the Widgets and a legitimate request loads just fine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=92) [Let's now change that CategoryId to an "x" again, reload the page, and we get a very different error message.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=99) [Now what we're seeing is still not very user friendly. This is what we would refer to as a yellow screen of death,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=107) [which is the same as what we'd call the previous screen except the previous yellow screen of death exposed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=114) [internal code implementation. So, we got the exception type, we got code lines, we got actual code syntax,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=119) [we got the ASP.NET and the .NET version -- we got a lot of information.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=127) [What we're getting this time is a very generic error message, which says, "Hey here's how to set up your](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=132) [Web.Config correctly." So this is much better, but it's still not real good.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=137) [Now one of the reasons it's not real good is that when this page loads, it's also returning an internal server](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=142)[error, HTTP code. Let's have a look at that. If I press F12 for the Developer Tools, Network, start capturing,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=149) [and we reload this page, here we'll see that we've got an HTTP 500.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=160) [Now one of the reasons that this isn't good is because there are a lot of automated scanners, so crawlers,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=165) [that will browse around the web trying to cause exceptions on web applications and they will look for HTTP 500s.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=171) [Now these crawlers are malicious and when they can find an application that's throwing HTTP 500s, that's a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=179) [red flag to an attacker who might then want to go and probe further and get this sort of information that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=186) [we got in the attack earlier. When customErrors is off, it also returns an HTTP 500.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=193) [So, again, it's like a little bit of a flag, which says "Hey here's an application that might have some](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=198) [security misconfiguration problems. You might want to go and probe a little bit further."](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=203) [So whilst we partially mitigated the risk, this is really still not quite right.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=208) [Let's jump back over to Visual Studio and we'll lock this down a bit further.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=212) [Now what we want to do next is going to help us in terms of security and it's also going to help us in terms](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=218) [of usability and that is -- we want to add a dedicated custom error page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=223) [So what I'm going to do is just jump over into my Solution Explorer and I am going to Add a New Item and it is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=229) [going to be a ‘Web Form Using a Master Page' and we will just call this guy Error.aspx.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=238) [And we'll use the default Site.Master master page. Now what I'm going to do is just put a little header in here](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=248) [and we'll say "Error has occurred." So this is a very, very generic error message.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=255) [Let's now jump back into the Web.config and what I'm going to do now with my customErrors is I am going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=262) [add in a defaultRedirect and that defaultRedirect is going to be Error.aspx.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=268) [So let's rebuild that and we'll jump back over to the application and let's see how it behaves a little bit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=276) [differently now. So, we're back on the last Error screen we saw, which is obviously still trying an exception,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=282) [but because we turned customErrors On, we're not getting those lines of code and other internal information;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=290) [we're just getting a very generic yellow screen of death. Now let's reload the application and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=295) [Okay, so this is a much better error and there are a few things that you'll note here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=301) [So first of all, ‘Error has occurred' with serving up that generic error page, which we just made and referred to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=306) [as the defaultRedirect path in the Web.config. The other thing that we'll see up here is that we have loaded](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=313) [Error.aspx and it's also passed a query string -- aspxerrorpath= the original path where the exception occurred.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=320) [So what actually happens here is that the application redirects to that error page and what I'd like to do is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=329) [just show you what that looks like in the Developer Tools again.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=335) [So before we do that, I'm just going to go back a page, now let's hit F12 for Developer Tools again, let's start](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=339) [capturing, and let's now pass through another "x", which will cause our exception, and so it happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=346) [So have a look at the network request pattern now. So we can see the request to the path with the "x" in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=353) [query string and that was just a get. Now what happened is the server responded with an HTTP 302, which is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=359) [a temporarily moved response and what that 302 does is it says go and load the error page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=366) [So what we've done is we've removed that HTTP 500 internal server error response code that we had just before,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=373) [so this is a much better pattern. Of course, it does still kind of disclose the fact that we are experiencing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=381) [an error in loading the error page though. Not so much because of the 302, but because the page that we've](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=387) [been redirected to adheres to this pattern where we've got the aspxerrorpath query string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=392) [Now this is something that some people don't like because of the fact that it really is disclosing that an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=398) [internal error has still occurred. So let's look at another way of configuring our customErrors](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=402) [and to do that, we'll jump back into Visual Studio.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=408) [So there's one more attribute on customErrors I'd like to have a look at and this attribute is called redirectMode.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=412) [Now redirectMode has got two options. The first one is ResponseRedirect and ResponseRedirect is what we're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=418) [seeing just before. So, the response comes back to the browser, it's a 302, the browser makes a request for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=426) [another page, which is the error page, and that's the ResponseRedirect pattern.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=433) [What we're going to look at now is the ResponseRewrite pattern and we'll see that that actually behaves quite](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=438) [differently. So, let's save that and we'll jump back over to the browser and see how that works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=443)[Okay, back in the browser, let's just go back to the Widgets page and we'll make a normal legitimate request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=451) [and then we will manipulate this query string parameter to an "x" again and load the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=458) [Now, the browser contents looks just the same as it did before when the redirectMode was ResponseRedirect.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=463) [Now that it's ResponseRewrite, the big difference is that the path hasn't changed, so we still see the original](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=471) [path that was requested -- Widgets.aspxCategoryId=x. So the URL is not disclosing the fact that an error has occurred.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=478) [The other thing is that if we open the Developer Tools again, so F12, ‘Start capturing', and we refresh this page,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=487) [we will see that it's an HTTP 200 response code. So this is a good configuration, in that what it's doing is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=495) [it is serving up a friendly error message to the user, which is much better than them seeing a yellow screen of death;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=502) [it's keeping that internal implementation data away, so we're not seeing any lines of code or syntax or framework](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=508) [versions, and it's also returning an HTTP 200 so we're not leaking via response header the fact that an error](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=514) [has occurred. We're also not leaking via the 302 redirect path, the one that had the aspxerrorpath](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=521) [query string in it, the fact that an error has occurred. So, in this configuration, we've really locked this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=528) [application down very well and the only way that you would know that an error has actually occurred is if you](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=534) [logically read the contents of the screen. Now, of course, a human can read that and say "Hey, it says an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=540) [error has occurred," but it does make it much harder for automated tools to use any sort of human logic to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=546) [figure out the fact that an error has occurred because all of the telltale signs, like response codes and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=552) [aspxerrorpath in the query string, no longer exist.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=4&mode=live&start=559)

[Demo: Securing web forms tracing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live)

[The next thing I'd like to look at is tracing in ASP.NET web forms.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=0) [Now tracing can be a very, very handy tool for debugging applications during development time or when](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=5) [troubleshooting later on. It can also be an extremely handy tool for an attacker because it discloses a lot](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=11) [of information. So let's have a look at what is in a trace.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=18) [And the first thing I want to do in my application is make a request so that we've got some data in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=22) [trace file, so light up the Widgets page, and now we will request the trace handler, which is at trace.axd.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=27) [And when we load that trace handler, we can see there's a few different requests here, the most recent will be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=36) [sitting up there in the top, so let's have a look at the details of this trace.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=40) [Now what we'll see is that there is quite a lot of information, which is disclosed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=45) [So, for example, we get the Session Id, we get some fairly generic information, which is not of too much risks,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=50) [things like the HTTP verb and the response code, but then we start to get a lot of information about the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=56) [internal implementation, so we can see each of the events in the page lifecycle.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=63) [One thing we can see here, which is extremely dangerous, is we can see that the developer in this case has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=68) [written out some debug information into the Trace Log and, in fact, what we're seeing here is a connection string.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=74) [Now this only exists because the developer has decided to put that connection string into the Trace Logs,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=81) [but certainly there are many cases where sensitive data is omitted into tracing for the purposes of debugging](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=87) [and then later not removed, so that's very risky data. If we scroll down a little bit further, we'll also](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=94) [get information like control trees, so this starts to give us a better idea of how the page is put together.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=100) [Now, of course, something like control trees, a lot of this can be gleaned from looking at the HTML source of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=107) [the page, but it does give an attacker just that little bit of extra information about the internal implementation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=112) [So, scrolling further down in the trace file, we will also see information around our headers and things like](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=119) [cookies are very sensitive and one thing that we saw in earlier modules is that cookies can often contain things,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=126) [such as Auth Tokens, and Auth Tokens can be used to hijack somebody else's session.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=131) [So certainly cookies is not something that we want to leave lying around for an attacker to come and pick up.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=137) [As we scroll down further, we'll also see information such as Forms and Querystring values.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=142) [So, obviously there could be some very sensitive information in there; very, very useful information to an attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=147) [And, of course, lots of Server Variables. So, in our Raw request data, we're going to get information again](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=153)[about cookies, we're going to get referrers, so where someone has come to the page from, and other internal](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=158) [information such as the physical path of where the page actually exists, remote address of the person who](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=164) [actually made the request, and other internal information, such as the version of the web server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=170) [Plus, of course, a little bit more information about the .NET framework version and the ASP.NET version.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=175) [So that's the same sort of info that we saw in that yellow screen of death a little bit earlier on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=180) [So the bottom line is - tracing has some extremely valuable information to an attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=185) [We do not want to have tracing left visible. Now fortunately tracing is very easy to turn off so let's jump](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=190) [back over to Visual Studio and configure that correctly.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=197) [So we're back in the Web.config of Visual Studio and tracing, as we can see here, has been set to enabled="true".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=201) [Now this is the problem -- if we do not have a tracing node at all, it will be false (false is the default) or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=210) [we can just go through and manually turn it on to false. Now once we disable tracing, we're going to see a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=216) [very different result when we request that trace handler. So let's jump back over to IE and see how that looks.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=222) [Okay, so we are on our trace handler, this is the request that we're just looking at.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=230) [Let's now try and reload that and we should get a very different response, and here we go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=234) [So, what we're actually seeing here is we're seeing that customErrors page, which we created earlier on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=240) [Now this is really important because it means that ASP.NET will not serve up the trace handler when it is disabled.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=245) [So very, very easy. All we need to do is make sure that we never get that trace handler enabled when we push](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=252) [the site through to a production web server and we'll look at some ways to make that easier a little later on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=5&mode=live&start=258)

[Demo: Keeping frameworks current with NuGet](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live)

[One of the things that OWASP talks about when they talk about security misconfiguration is the importance of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=0) [keeping frameworks up-to-date and when we talk about frameworks in the context of ASP.NET, obviously there are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=6) [the .NET frameworks, but there are also third-party libraries.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=11) [And particularly given most web applications today do have quite a large collection of third-party libraries](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=15) [in them, it's very, very easy for libraries to get out-of-date and in some cases, those libraries have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=20) [security vulnerabilities so we need an easy way of keeping track of how libraries are changing and when they](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=26) [want to, being able to take an update into their application in a very low-friction way.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=32) [Now the best way that we have to do that in ASP.NET and Visual Studio today is by using NuGet and, in fact,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=37) [every time you create a new project in Visual Studio 2012, you get a whole bunch of NuGet packages in there](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=44) [automatically. So let's take a look at how NuGet is configured in this almost entirely standard ASP.NET](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=51) [web forms application. So, what we're going to do is just right-click on our References and go to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=58) [Manage NuGet Packages. Now, what NuGet is doing is jumping off to the internet and having a look at what](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=63) [packages are available for update, but before we do that, I want to have a look at our Installed packages.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=70) [And what we'll see here is that there are actually several pages worth of packages already installed in this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=75) [ASP.NET application. Now I added AntiXSS myself and we'll come back to that in a moment, but you'll see that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=80) [there are a whole bunch of other packages, so things around OpenAuth, there's entity framework built in there;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=86) [we've got jQuery - jQuery is a good example of a library which updates quite frequently;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=92) [we've got Universal Providers for memberships; we've got friendly URLs; we've got web optimization; and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=97) [we've got things like Modernizr and WebGrease for helping us do bundling and minification.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=103) [So there are a lot of packages already built in to an ASP.NET web forms application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=108) [Now one of the great things about NuGet is that it's very easy to update any of these packages.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=113) [So, if we jump over into the Update menu, we will see all the packages which are due for an update and,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=118) [in fact, we can see quite a lot already. So right out of the box, because this template was built some months ago,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=124) [there are already updates, but what I want to look at though is AntiXSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=131) [So let's take a look at what we're seeing here. Now, first of all this is a Microsoft Package.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=136) [What we're seeing here is that the installed version is 4.0.0, but there is a 4.2.1 available.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=140) [Now I want to put this back in a security context and have a look at a bit of information about 4.0.0.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=148) [So let's jump back over to Internet Explorer and I'll help put the security side of this in context.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=155) [So here's the problem we had with AntiXSS. There was a vulnerability that was identified in early 2012 and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=161) [the risk in this case is that there was the potential for information disclosure, which is obviously not](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=168) [something that we want in our applications. Now this security update was rated "Important" so certainly it's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=174) [something that you would want to take seriously if you're using the framework and, as we'll see here,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=182) [it affected AntiXSS V3 and it also affected AntiXSS V4.0. Now we just saw that we were running 4.0.0.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=187) [So, clearly what we want to do is update our application to patch this vulnerable version of the AntiXSS library.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=197) [So let's jump back over to Visual Studio and we will upgrade it in a single click.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=205) [So we're back in Visual Studio and now we know that there is a bit of a risk in this particular version of AntiXSS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=211) [The great thing about NuGet is in order to fix this risk, it is just simply a matter of clicking the Update button.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=217) [We may get some licensing information like so, but we'll just accept that, and NuGet is now going after the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=223) [internet, it has actually pulled the package back, and installed it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=229) [So now we've got a little tick next to it and the package has been successfully updated.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=233) [So that's the great thing about NuGet; it really is a very, very simple, often one-click affair to update](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=238) [vulnerable packages. Compare this in the past to having to go and seek out the information about packages](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=244) [that have changed, find the right version, manually extract it, include it into the application, also apply](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=250) [any configurations that need to be made, such as Web.config settings -- NuGet does all of that automatically.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=256) [So it really, really helps with that particular risk in security misconfiguration about keeping libraries and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=262) [packages up-to-date.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=6&mode=live&start=268.5)

[Demo: Encrypting sensitive parts of the web.config](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live)

[One thing we frequently need to do when we're building ASP.NET applications is store sensitive information in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=0) [the app. Now, in this case, I'm talking about configuration data. So, for example, we often need to store](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=5.5) [a connectionString in the application, the app frequently needs to have the credentials required in order](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=12) [to connect to the SQL Server. Now, earlier on we saw how that connectionString was actually leaked through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=16) [a security misconfiguration risk. The connectionString existed in the code and due to misconfiguration of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=23) [customErrors and poor error handling, it was leaked externally.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=29) [Now obviously an attacker can do some serious damage with that information.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=32) [What we want to do is make sure that we keep things like connection strings in our Web.config and, indeed,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=37) [there is a dedicated connectionString section in the Web.config.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=43) [However, we still don't want to have sensitive data floating around in plain text in the Web.config.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=47) [So, for example, here we can see that same connectionString in plain text.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=54) [Now whilst this should not be able to be disclosed in a live application, there are other risks.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=59) [So, for example, a very typical risk is we want to put this application under source control.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=65) [Now we don't necessarily want every person who has access to the source control repository to see our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=71) [production connectionString details so we need a way of properly securing them and we do have a very good](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=77) [native way within ASP.NET and IIS, which makes it very, very simple to encrypt this connectionString.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=83) [So let's take a look at what's involved in doing that. Now to do that, we need to get a command prompt as Admin.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=91) [So in Windows 8, it's very easy, just Windows x key and then choose ‘Command Prompt (Admin)' or in Windows 7,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=96) [just hit the Windows key, type command prompt, right-click on the command prompt option, and choose Run](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=102) [as administrator. Now, when that command prompt opens up, we should see that it is running as administrator,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=109) [which is a very good start. Now in order to encrypt the connectionString, we are going to run](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=115) [the aspnet\_regiis command and where we want to get that command from is in Windows and then in Microsoft.NET,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=120) [in Framework, and then it will be in your version of .NET, which in this case is v4.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=130) [Now to actually encrypt the connectionString is very simple. So, we're going to run aspnet\_regiis and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=137) [we're going to pass a site parameter and the site will be the name of the site in IIS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=145) [In my case, it is 6-SecurityMisconfig. We then need to specify the name of the app in that website and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=150) [in this particular case, the app is at the root of the website, so we can just put a slash, and then we will](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=159) [use the "pe" switch to specify the section of the Web.config that we want to encrypt.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=165) [Now in this case, it's just going to be the connectionStrings and that should do it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=171) [So let's run that command and we can see here, Encrypting configuration section succeeded.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=176) [Let's jump back into Visual Studio and now have a look at what our Web.config looks like.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=183) [So Yes to reload that and have a look at this -- it is a very, very different connectionStrings node.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=187) [Now what's actually happened is that aspnet\_regiis has used the RSA encryption key stored on the server](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=193) [in order to implement symmetric encryption. So even though what we're seeing here now is a cipher,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=201) [when the application actually executes, this will be decrypted and everything will run just fine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=206) [Now if we need to, we can also jump back to the command prompt and we can decrypt and decrypt is simply done](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=212) [by changing that "pe" switch to "pd", Run, decrypting configuration section has succeeded; now let's have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=219) [another look at that connectionString. So, reload and we're back to normal.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=226) [So that makes it extremely easy. Now we could put that encrypted connectionString into source control or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=231) [share it with other developers and unless they are actually able to run it on the server, they won't be able](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=237) [to get that connectionString back to plain text. Now you can do this with other areas of the Web.config as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=243) [So you can secure other sensitive data that you may store in the file.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=248) [So this is a really, really good way of making sure that you're not leaking any sensitive information to your](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=252) [own development team and, of course, the other thing is, is that if an attacker does get hold of the Web.config,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=258) [they can't do anything with it either. If they do actually manage to get remote access to the box and they](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=264) [can go on and run the Decrypt command, then yes -- that would be a risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=270) [So certainly this is not foolproof, but it does offer a far greater level of protection than just having it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=275) [sitting there in the Web.config in plain text.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=7&mode=live&start=281)

[Demo: Using config transforms to apply secure configurations](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live)

[One of the real problems with security misconfiguration is it's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=0) [just so easy to get one little configuration setting wrong and have it find its way into a production environment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=3) [It's very easy, for example, to accidentally leave tracing on or to turn customErrors off just for that little](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=10) [while while you test something and then not remember to turn it back on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=15) [Certainly I've seen many, many cases where developers know better, but it's just been one little oversight and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=19) [a risk has slipped through by virtue of security misconfiguration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=26) [Often this risk exists because the deployment process hasn't correctly configured the application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=30) [Now one of the defenses that we have at our disposal is called config transforms and config transforms give us](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=36) [the ability to adapt the Web.config based on the build configuration of the application when we deploy the app.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=43) [So, what you will find in any brand-new ASP.NET application is under the Web.config in your Solution Explorer,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=51) [there will be a Web.Debug.Config and a Web.Release.Config. Now let's take a look inside that Web.Release.Config](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=59) [and what you're seeing here is what we refer to as a config transform.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=68) [Now what the transform does is it describes how the Web.config should be changed during deployment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=72) [So, for example, one thing it says is that in system.Web, the compilation node should undergo a transformation,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=79) [which removes the attribute called debug. So what this means is that when we don't have a debug attribute on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=88) [compilation, the application will fall through to the default, which is debugging being off.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=95) [So, if we publish this application using the Release.Config because we've got to remember each of these](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=100) [config transforms relate to a build configuration, so in this case we have the debug or release, the application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=106) [will automatically have the debug attribute removed from compilation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=113) [You'll also see that there's a little bit of configuration here that's commented out for customErrors.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=117) [So, if we were to remove this commenting here, one of the things we'll see is that customErrors will be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=122) [automatically set so there's a defaultRedirect of GenericError.htm, but the mode is set to RemoteOnly and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=129) [the Transform is replaced. So it's going to replace the existing defaultRedirect value and it's going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=136) [replace the mode value. Now we've actually added our own defaultRedirect so I'm going to remove this and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=142) [we might just leave the mode set to RemoteOnly so that we can see this change.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=148) [The other thing it's going to do is it's going to add a dedicated redirect path for an HTTP 500 statusCode, so that's fine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=153) [The other thing I'd like to do though is a little bit earlier on we looked at tracing and the risk of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=160)[trace handler being publicly accessible in the web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=166) [So let's use the config transform to make sure that that trace handler is never accessible when we publish for release.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=169) [So, what we're going to do is just add a trace node and the Transform that I want to use in this case is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=177) [going to be a "Remove" Transform so I actually want to just delete that node altogether.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=183) [When trace doesn't exist, it will default to being disabled.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=189) [So, that's great. This is the way we want to securely configure our release configuration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=193) [Now to actually apply the configuration, we need to publish. Just saving this or copying and pasting the files](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=198) [over won't actually apply it; it has to go through a transformation process and that's what happens during Publish.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=204) [So let's jump over to the Solution Explorer, right-click, and then we'll jump into Publish.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=211) [So, here is our default Publish window and what we're going to need to do here is, first of all, set up a profile](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=217) [and all we really need to do with this profile is give it a name.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=223) [So we will just call it ‘Deploy to file system' because what we're going to do is actually publish this to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=225) [my local file system and then take a look at how the Web.config has changed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=231) [So, the Publish method that we're going to use is File System, you'll see that we could FTP it or we could](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=235) [use Web Deploy; we're just going to keep it very simple for the sake of config transform demos and what I will do](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=241) [is just put it straight over in my Temp directory and we'll hit the Publish button.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=247) [Okay, so that is done. Let's now take a look at what's in the Temp directory.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=254) [So, here is the Temp directory and this looks just like a normal published web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=259) [There are no code-behinds, they've all been compiled into the bin, and of course we have a Web.config file.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=265) [So let's take a look inside here and we'll open that up in Notepad and there will be a couple of things to note.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=271) [So first of all, customErrors had the mode set to RemoteOnly and we can also see that new statusCode has been added.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=279) [The other thing is, is that there is no trace file left. So if we jump back into Visual Studio, we will see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=287) [that we had tracing in the Web.config and it was sitting there just between customErrors and compilation and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=292) [when we look back at the transformed file, there is nothing there between customErrors and compilation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=299) [So trace has been successfully removed. So, config transforms are a very powerful feature because it allows](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=304) [us to configure our Web.config securely through our Release.config transform and then every time we do a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=311) [deployment that secure configuration will be applied. So long as nobody then goes and circumvents the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=318) [deployment process, so for example by editing the Web.config directly on the target server, you can always](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=325) [have confidence that every deployment will be done with your secure configuration.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=8&mode=live&start=332)

[Demo: Enabling retail mode on the server](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live)

[The last thing I'd like to look at in this module is a configuration called retail mode, which is a configuration](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=0) [that we're going to actually make on the machine. Now what retail mode does is it means that even if we get](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=6) [our security configuration wrong and we experience exceptions, such as you see in front of you now, that we don't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=12.5) [actually get messages such as you see in front of you now. We don't want to get that internal exception with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=18) [leakage of code. The only reason we're seeing this now is that I've reverted the good customError configuration,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=25) [which we made earlier on, because I want to show you how it changes when we enable retail mode.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=31) [Now to enable retail mode, I'm going to open up the Machine.config of this particular PC and it would be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=37) [the same case if you were doing this on a server and in order to edit that Machine.config, I'm going to need](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=44) [to do so as an administrator. So I want to open up Notepad as an administrator.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=50) [So I'm on Windows 8, I'm just going to hit the Windows key and type notepad and then I'm going to right-click](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=54) [on Notepad and choose Run as Administrator.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=61) [Let's now open up the Machine.config and this is the path we're going to open it up from.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=66) [So I was in Windows, Microsoft.NET -- in my case it's a 64-bit machine so it's Framework 64, the version of .NET](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=73) [and config, and then the filename inside there is Machine.config. And here it is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=80) [Now what I want to do is just scroll down to the bottom of this Machine.config to where we have the System.web element.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=86) [Now enabling retail mode is actually very simple and all it means is that we will add an element anywhere in here](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=93) [and we'll just call it deployment and retail="true" and because we open this as an administrator, we should be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=101) [able to save this file; you won't be able to save it if you don't open it as an admin.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=111) [So let's now jump back into the app and give it a refresh and we should see a very different result.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=115) [And here we go. So we're back to the point where we were before when we configured customErrors when all we did](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=121) [was turn it on, so there was no defaultRedirect page. So this was not our ideal configuration, but it has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=126) [immediately stopped the application from exposing the internal implementation of the code.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=133) [So this is a very good safety net and what it means is, is that every application running on this machine,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=139) [at most, will display the yellow screen of death you see in front of you.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=146) [It won't display the one with lines of code and internal exceptions and Framework versions and everything like that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=149) [So it has entirely disabled that feature. Now this can also make things difficult at some times.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=156) [So, for example, if you really want to be able to see what that internal error message is, you will not be able](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=162) [to do it whilst retail mode is on, but of course that's not really something that you should be doing in a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=168) [production customer facing environment anyway. You're far better off using a tool, such as ELMAH,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=174) [for error logging so that you can go back and review it in a more secure fashion later on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=180) [So retail mode isn't for everyone, but it is a way of ensuring that you don't accidentally let an important](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=185) [security misconfiguration slip through into a production environment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=9&mode=live&start=191)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live)

[Let's summarize the module and I think probably the most important thing to take away from it is that it is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=0) [very, very easy to make a very, very small change, particularly to the Web.config, and introduce the risk of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=5.5) [security misconfiguration. Simply turning custom errors off or enabling trace is enough to leak serious data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=13.5) [about the internal implementation of the application. And as we saw with custom errors, that data could at times](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=21.5) [be used to immediately breach the database. So, again, very, very simple configuration change -- a potentially](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=27.5) [major adverse outcome as a result. Keeping those packages up-to-date.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=34.5) [You need to have some sort of a plan for how you're going to keep frameworks, libraries, packages, whatever you'd](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=39.5) [like to call them, up-to-date and NuGet is the one that makes the most sense for ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=45.5) [Now NuGet is a great thing not just for security, but for realizing new features, for taking other bug fixes,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=51.5) [basically getting anything new from the developers of the package. The fact that in the case of something like](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=59.5)[AntiXSS, it helps us take a new library with a previous vulnerability patched, that's a very positive thing for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=64.5) [security misconfiguration. Remember to protect anything sensitive in your web.config.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=70.5) [We looked at how we can very easily encrypt sections of the Web.config using aspnet\_regiis.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=77.5) [Always try and keep in mind that anything in your web.config may be visible to other people.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=84.5) [Now hopefully that's a very small audience, but certainly there are many precedents of say web.configs with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=90.5) [connection strings being placed on the GitHub and being accessed by a much broader audience.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=96.5) [So keep that in mind and keep that data encrypted. Also be conscious that very often security misconfiguration](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=101.5) [comes about as part of the deployment process. So, particularly in web.config, there are many, many instances](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=109.5) [where configuring the web.config for a production deployment is a manual process.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=116.5) [People will publish the app and then go through and manually edit the settings, such as connection strings,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=122.5) [or do so and manually copy and paste things. Now every time we do something manually, there is a risk of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=127.5) [getting it wrong and in many cases, particularly for web.config, that could be a security misconfiguration risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=132.5) [So use config transforms to automatically do those deployments. The config transforms go into source control,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=139.5) [they can be applied automatically on Publish, it's a great way of mitigating this risk and just making your life](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=146.5) [generally easier when it comes to deploying an application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=152.5) [And lastly, remember there's that retail mode safety net. So if you really do want to make sure there's just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=155.5) [no possible way you can have a serious custom errors misconfigurations, turning retail mode on the server](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=161.5) [is a good way of doing that. Keep in mind also that it means you're never going to be able to see an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=168.5) [internal error message in that environment, which really is a good thing, but it might also mean some other](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=172.5) [changes to your processers needed if you relied on that in the past.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=179.5) [So have a look at something like ELMAH as a way of securely logging those errors and accessing them at a later date.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=184.5) [And finally, remember that security misconfiguration is a very broad generic risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=190.5) [These are just a few of the items that you need to think about in ASP.NET, but it's certainly not exhaustive either.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m6-security-misconfiguration&clip=10&mode=live&start=195.5)

[Insecure Cryptographic Storage](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to Part 7 of the OWASP Top 10 on Insecure Cryptographic Storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=0) [In this module, we're going to start out by taking a look at how OWASP views the risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=8) [We're then going to move on to performing an attack against weak password storage, so we're actually going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=13)[crack some passwords here. Now once we've done that, we'll work on better understanding password storage and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=19) [what it means to hash a password. We'll also take a look at how salt can be used to help protect passwords](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=25) [and we'll have a look at where salt can still be vulnerable. Certainly just salting hashes is not enough to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=32) [properly protect passwords. So after that, we will take a look at how we can create stronger hashes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=37) [Now beyond just password storage, we'll also take a look at asymmetric encryption, symmetric encryption, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=44) [we'll also take a look at the data protection API or DPAPI as it's commonly known.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=50) [And finally, we'll also take a look at a few myths and misconceptions about cryptographic storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=56) [Certainly there's a few ideas out there about how data can be secured, which really isn't very secure at all.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=0&mode=live&start=62)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live)

[Moving on to the OWASP overview and risk rating, once again the risk is pretty much anyone who can access](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=0) [the system and in the case of cryptographic storage, it's people who might try to gain access to data that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=6) [they're not meant to gain access to, so data that is protected cryptographically.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=12) [And, of course, this may also include internal administrators. So, one of the purposes of cryptographic storage](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=17) [is to provide protection from people who do actually have access to the data, but we may not want reading it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=22) [Exploitability is classified as difficult because when we talk about cryptographic storage, we're not really](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=31) [talking about breaking algorithms, we're talking about breaking some other aspect of the way that data is stored,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=37) [such as gaining access to private keys or as we'll see shortly, brute-forcing hashed passwords and sometimes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=44) [that actually can be rather easy. The prevalence has been marked as uncommon and the detectability difficult](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=51) [simply because from an external perspective, it's actually difficult to discover how data has been protected](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=58) [internally within the system. There are some giveaways, which we'll take a look at, but regardless there still](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=65) [needs to be some other part of the system exploited before an attacker can actually get their hands on encrypted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=70) [and circumvent the cryptography. So really, the risk of insecure cryptographic storage is a risk, which is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=77) [only realized after another risk has already been exploited.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=83) [Now, in this case, the Technical Impact is flagged as severe simply because normally when data is protected](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=88) [by some means of cryptographic storage, that data usually has a high value.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=94) [So often it's very sensitive data or its data which may cause serious damage if an attacker gains access to it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=99) [So consequently, if they are actually able to circumvent the cryptography, the likelihood of the impact being](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=106) [severe is very high. Now, of course, the Business Impact always depends on the nature of the data,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=112) [but again particularly when we talk about cryptography, if there was an attempt to protect that data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=118) [and the cryptography simply wasn't sufficient, the chances are that exposure of that data is going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=124) [to have some pretty serious business impacts.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=1&mode=live&start=129)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live)

[To actually demonstrate the risk of insecure cryptographic storage in practice, I'd like to take a look at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=0) [breaking passwords stored by the ASP.NET membership provider.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=7) [Now this is a very common way of providing registration and log-in facilities simply because it's so easy](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=11) [to provision within a brand-new ASP.NET project. Now there have been some changes to the way credentials](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=18) [are stored and we will look at those changes later on. So what I'm going to do is I'm actually going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=25) [start with an old template and what we're looking at here is we're looking at a Visual Studio 2010 ASP.NET](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=30) [web forms application. Now what I've done here is written a little script to automatically load passwords into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=37) [the membership provider and, in fact, you'll see here I have a little page, which is called LoadSonyPasswords,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=44) [and what this script is going to do is it's going to go and grab all the passwords in a file called](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=52) ["Sony passwords.txt". Now in 2011, Sony had a large number of breaches against different systems and in some](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=57) [cases, the passwords were stored in plain text. Now what that means is, is that we have a very good record](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=66) [of the sort of typical passwords that users choose. So, what I've done here is I've saved 20,000 of those](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=72) [passwords into a file called Sony passwords.txt and as you'll see if I scroll through here, these are very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=79) [typical-looking passwords, they're predominantly lowercase, they're quite short, some of them have numbers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=86) [as well, and it goes all the way down to 20,000 records. So, what's going to happen is this page automatically](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=91) [enumerates through every password in that file and it calls into the membership provider and it creates a new user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=99) [Now it just increments the user id, so it'll be user 1, 2, 3, all the way through to 20,000, and it just sets](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=106) [the same email address on all of them, but what's really important is that it's going to pass through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=113) [a different password. Now I've already run this script so let's jump over to SQL Server and have a look at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=118) [what we now have in the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=123) [So when we use the ASP.NET membership provider FROM, and we did have a look at this earlier on in Part 3 on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=126) [Broken Authentication and Session Management, all of the user accounts are loaded into a table called](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=132) [aspnet\_Membership. Now if we go and execute this query, we should see that in our table we have a whole bunch](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=138) [of records, which have a password, and clearly this password has been stored cryptographically.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=145) [We also have a password salt column and we'll talk a little bit more about salts later on, but these two](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=150) [columns are the important ones for this exercise. If we scroll through all of these records, we should see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=156) [that we have a whole heap of them all the way through to 20,000 separate records.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=161) [So this is a very, very typical set of passwords because it's come from real-world passwords used at Sony.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=166) [Now what I want to do is save these passwords and password salts in a certain format and put them in a text file](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=173) [and then we'll move on to the cracking process.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=178) [So I have another little script over here and what this script does is it selects out the password salt and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=181) [the password and it surrounds it with a certain pattern and we'll have a look at why this pattern is important](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=187) [in a moment. The main thing is, is that when we execute it, we will get 20,000 rows and they will all look](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=192) [kind of like this. So earlier on I've gone and saved these into a text file.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=199) [So let's jump over to Windows Explorer and we'll take a look at the next part of this process.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=203) [So here we are on my C: drive and we're in a folder called oclHashcat-plus.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=209) [Now, Hashcat is a hash-cracking tool and this is what we're going to use to break these passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=214) [It's a free tool designed for "password recovery," but of course it's also very convenient for attackers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=221) [Now what you'll see in here is that we have a file called MembershipAccounts.txt and this has simply stored](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=228) [20,000 records selected straight out of SQL Server using that pattern that we just looked at.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=235) [Now that last thing I want to show you before we actually start cracking passwords is that in here, we also](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=240) [have a filed called hashkiller.com.dic. Now what this is, is a collection of passwords in what we refer to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=245) [as a password dictionary. Now what a password dictionary is, is it's not like an Oxford or a Cambridge or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=254) [something with normal words, but it's a collection of common passwords that people use often from other breaches.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=259) [So when we look at a word like this, obviously it's not really a word, it's a combination of letters and numbers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=267)[and punctuation, and in this particular dictionary, it goes all the way through to over 23 million records.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=272) [So what Hashcat is going to do is it's going to use this dictionary and it's going to hash the passwords](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=280) [and then it's going to see if that matches anything from our breach and if it does, it will have been able](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=285) [to work out what the plain text version of the hashed password is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=290) [So what we're going to do now is drag over a Command window and in this Command window, we are going to run](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=294) [a command which will take the passwords that we saved out of SQL Server, it will take the password dictionary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=300) [we just looked at, and it will try and use that dictionary to crack the passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=307) [And the command will simply look like this. Now this is very simple, there's just a switch for the password pattern](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=311) [and in this case, the pattern that the ASP.NET membership provider used in the Visual Studio 2010 template is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=318) [just a salt that's combined with the password and hashed using the SHA-1 algorithm, we're passing in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=325) [MembershipAccounts, which we just saved from SQL Server, and we're giving it the hashkiller dictionary.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=330) [So let's run this and we will see what happens. Now, we can see right at the top, we have found 20,000 hashes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=335) [and there are 20,000 unique salts and as a result, there are 20,000 unique digests.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=343) [We can also see down here that we have 23 million-something passwords from the dictionary and what we'll now](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=350) [start to see is that row by row, Hashcat is going through and cracking passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=356) [So what we see on the left of the screen is the password and the salt that we saved from the database and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=364) [everything on the right-hand side is the plain text password. So, even though we have a salted hash,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=371) [we can see that Hashcat is now moving through and cracking all of those passwords one-by-one and it's actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=377) [doing it very quickly. So let's leave this to run for a little while and then we'll come back to it right at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=383) [the end of the process.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=389) [We're just getting to the very end of the hash set now and here we go -- it has completed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=396) [So let's have a look at what Hashcat has managed to do. And the first thing I want to point out is the speed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=401) [So down here, we can see that there have been over 70,000 hashes computed per second, which sounds like a lot,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=407) [but in fact it's a very, very small percentage of what this GPU can actually do and we'll talk a little bit more](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=414) [about why this is using the GPU and not the CPU just a little bit later on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=421) [Now the next thing we'll see on the line below is that over 13,000 passwords out of the 20,000 password sets](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=426) [have been recovered so that's about 65% of passwords even though the restored as salted hashes have now been cracked.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=433) [So that is a significant number and that number would be significantly higher if we had a larger password](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=443) [dictionary or if we spent more time mutating the passwords, so doing things like character substitution,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=448) [what if we change the a's for an @ symbol or if we changed the casing.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=455) [There are lots of different ways an attacker will mutate input in order to try and crack different passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=459) [And the final thing worth noting down here is the time -- so that was only about 14 minutes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=466) [So we just cracked 13,000 salted hashed passwords, 65% of the passwords set in only 14 minutes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=472) [Now this was also done with consumer-level hardware and we'll take a look at what that looks like later on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=480) [in the module. Now the final thing is, let's take a look at the output of the whole process because all of this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=486) [has been saved into a file, which we can now load back up, match to the hashes in the database, and figure out](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=492) [which usernames had which passwords. So this whole process created a file called hashcat.pot and if we have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=499) [a look inside this file, what we will find is that we have our patent, which we fed in in the first place, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=507) [then we have the plain text password after this. So clearly we could now go back to the database and say](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=514) [for this password and this password salt, which of course will uniquely tie us back to the account, this is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=520) [the plain text password and you can see, as we scroll through this list, these passwords have all sorts of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=526) [different formats. They're not just simply incrementing numbers or just a lowercase character space;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=532) [they're very, very random because we had the password dictionary and that password dictionary is a really good](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=539) [representation of the actual passwords that people create. So, in short, what this shows us is that the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=545) [password storage in the membership provider in a Visual Studio 2010 template for a web project is next to useless.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=552) [This is just far too many passwords cracked far too quickly. So let's now move on to having a look at how we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=562) [should do our password storage and how we can make it significantly more secure than what we've just seen here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=2&mode=live&start=568)

[Risk in practice: ABC passwords](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live)

[As usual, I'd like to take a look at an industry precedent of insecure cryptographic storage and this time,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=0) [I'm going to use an example, which I wrote about at the start of 2013.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=6.5) [Now what happened in this case is that the ABC, which is the Australian Broadcasting Corporation, had a website](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=11) [breach, which exposed about 50,000 passwords, which were stored as a SHA-1 hash.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=18) [Now the first thing to point out here is that there was a vulnerability in the website, which allowed the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=25) [attackers to actually extract those hashes from the database. So breaking the cryptographic storage was really](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=29) [only something that could happen after the data had already been exposed by virtue of another vulnerability](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=36) [in the website. Now as often happens, once an attacker gains access to that data, it very, very quickly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=43) [ended up public and we particularly see this with that hacktivist group of attackers where they want to share](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=49) [their exploits and publish the data that they've been able to extract from the victim.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=55) [Now in this case, once said data was public, I wrote a piece about how easy it was to crack it and, in fact,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=60) [what we'll see here is that I've used Hashcat and Hashcat has been able to go through and recover 18,406 of the hashes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=66) [Now these were just SHA-1 hashes with no salt and what I wanted to do was just demonstrate how fast it is to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=76) [break these hashes so it was only 45 seconds. So this is a really good example of insecure cryptographic storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=83) [So the developers did apply cryptography insofar as they used a hashing algorithm, but clearly if you can crack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=91) [18.5 thousand hashes in only 45 seconds, it is a very insecure mechanism of cryptographic storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=99) [Now this is just one example, but the same scenario has played out many, many, many times over recent years,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=108) [where passwords have been dumped publicly and the cryptographic storage has been so insecure that it has been](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=115) [extremely easy to go through and crack it. What we need to remember is that cryptographic storage is really](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=121) [the last line of defense; that's the one thing that saves the data once it has already been exposed by other](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=128) [risks in the application. So it's essential that cryptographic storage is done right because it's the very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=134) [last chance you have to save that data from being exposed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=3&mode=live&start=140)

[Understanding password storage and hashing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live)

[Let's start to gain a better understanding about password storage practices and the first thing is,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=0) [is that when we talk about cryptography. Very, very frequently, particularly in web applications, we're talking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=5.5) [about protecting credentials. Now certainly there are other classes of data that are protected in web applications,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=12) [like credit card data for example, but that's a very, very small percentage compared to the number of websites](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=18) [that need to store credentials. So getting password storage right is extremely important because it is just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=24) [so prevalent. Now we normally see three different ways of storing passwords and the first way is just in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=31) [plain text with no cryptography. So clearly if passwords are stored only as plain text and the application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=38)[gets breached, there is no further line of defense. That data is exposed and it's game over in terms of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=44) [an attacker getting access to the data. The other method of password storage is to use encryption and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=51) [by encryption, we are talking about an algorithm which normally has 1 or 2 keys and we'll look at the difference](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=58) [shortly and we're looking at the ability to both encrypt and decrypt.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=63) [So it's important to remember that encryption is a two-way process.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=68)[And the third way is to hash passwords, which is a very different process, and that's what we saw earlier on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=72) [in terms of what we broke in the membership provider and it's also what we saw earlier on in that real-world](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=79) [example of the way the ABC stored their passwords. So let's move on and take a look at the difference between](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=83) [encryption and hashing.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=88) [Now the first thing I want to make really clear here is that hashing is not encryption.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=91)[Both hashing and encryption are cryptographic processes, but they are fundamentally different,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=96) [and the main difference is encryption is a reversible process.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=103) [Now normally when we talk about encryption of passwords, such as something that happens internally within the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=107) [system, their tends to be a single private key and no public key, so this is what we would refer to as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=112) [symmetric encryption. Now the problem with encryption is decryption and the reason that is a problem is that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=118) [if a system is breached and an attacker can gain access to the data, they can also very often gain access to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=125) [the encryption key. Once that encryption key is obtained, then the cryptography used to store the passwords](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=132) [is next to useless, it will be broken. Hashing is a very different process because hashing is one way.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=139) [So you can hash, but you cannot unhash; there is no key involved in a hash.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=147) [Now the reason that's important for password storage is it means that even if the entire system is breached,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=153) [so not just the passwords, but the algorithm, the web application, any configuration settings, an attacker](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=159) [can't immediately gain access to the plain text. Now, of course, we can have insecure hashes and that's what](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=166) [we just looked at, but certainly there's not a facility for an attacker to simply put in a key and get the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=172) [plain text out. So the trick for us now is to look at how we can properly secure our hashes so that even when](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=177) [an attacker does get access to them, it's not going to be easy to get the plain text back out.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=184)[Let's take a closer look at hashing because it might sound a little bit odd to some people that you can't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=190) [actually unhash a password. So the first thing to understand about hashing is that it's keyless, which we've](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=195) [already discussed. It's also one way so there's no unhashing, we also touched on that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=201) [But the third thing is, is that a hash is a deterministic algorithm.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=206) [Now what that means is that every time we use the same algorithm to hash the same piece of plain text,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=210) [it produces the same cipher, so it produces the same output. What that means is, is that every time that piece](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=217) [of plain text is entered, there is a process we can use to produce the same output.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=224) [So it means that when an account is created, we can take that plain text, we can hash it, and we can store it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=230) [Now when the user comes to log in again, we can take the password that they provide at the point of login,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=236) [apply the same hashing algorithm, and compare it to the one in storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=242) [So you're never actually unhashing the password; you're never comparing the plain text password that was](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=246) [provided at registration with the one that was provided at login; you're only ever comparing the hashed versions](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=252) [of both of them. So that's really important because it means that the system never deals with plain text after](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=259) [it is provided either at the point of registration or the point of login, it's just immediately hashed and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=265) [either stored or in the case of login, compared to the one in storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=271) [So that's quite a secure way of dealing with this sensitive piece of data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=276) [One thing that was mentioned in the overview at the start of this module is that cryptographic algorithms](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=281) [themselves are not normally cracked. They're circumvented by other means, such as disclosure of the key or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=286) [in the case of hashing, by brute force. So in the example we just looked at with the ASP.NET membership provider,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=293) [that was a brute force attack. So that was Hashcat enumerating through a huge number of passwords in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=300) [dictionary of sample passwords, hashing them, and comparing them to the passwords that we downloaded from the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=306) [database, the ones which were saved by the membership provider.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=313) [The success of a brute force attack like that is very dependent on how fast those hashes can be recalculated.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=317) [In that example earlier on we saw something like 70,000 hashes per second were being calculated and at the time,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=324) [I said that's actually quite a small number. In reality, this can happen billions of times per second on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=330) [consumer-grade hardware and we're going to look at that in more detail in just a moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=336) [The important thing for now is to remember that the way these hashes get cracked is by speed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=341) [The faster the hashes can be recalculated, the faster an attacker can compare them to a breach.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=347) [Now none of this is about being 100% secure; that's not something that we ever look at as being a feasible](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=352) [option in security. It's all about how can we make this difficult enough that it no longer becomes feasible](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=359) [or financially viable for an attacker to bother and ultimately that's what a lot of security is about --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=366) [making it hard enough, not aiming for this impossibility of 100% secure, but aiming to make it hard enough](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=372) [for an attacker to break the system. To give an example of just how many hashes can be computed per second,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=379) [let's jump over and have a look at the Hashcat page. Now this is the tool, which we just used to break those](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=386) [membership provider hashes, and one of the things we see about Hashcat is that first of all it supports a number](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=392) [of different hashing algorithms. So, algorithms like MD5 are very common, the SHA variance are very common,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=397) [and in fact it was SHA-1 that we saw in that ABC example earlier on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=405) [Now what's important is that as we scroll down to the performance, if we take for example a Windows 7 PC 64-bit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=409) [and most importantly using an AMD hd7970 graphical processing unit or GPU and this is a piece of consumer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=417) [hardware that costs several hundred dollars, what we can see here is that for that PC1, if it is cracking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=426) [NTLM hashes, it can actually do about 7.5 thousand million or in other words, 7.5 billion computations per second,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=432) [MD5 about 5.5 billion, SHA-1 just over 2 billion. Now that is a huge number of hashes per second.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=442) [The reason why the example I used earlier on against the membership provider was only that 70,000 per second](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=451) [is I was feeding that in from a password dictionary so it actually had to pull this data off the file system](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=457) [and feed it into the GPU. The figures that you see on the page here are what can be achieved when Hashcat](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=463) [uses what's referred to as mutations. I mentioned mutations earlier on -- it's things like substituting A's for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=469) [@'s or maybe L's for 1's and that's a process that can happen within the GPU so there's not an I/O limit in terms](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=476) [of the amount of data that can get fed into it. So, what that would've meant if we had have run this process](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=484) [is that the potential passwords tested would've been a significantly greater range because it would've taken](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=490) [those 23 million-odd passwords from the dictionary and multiplied that out many, many times.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=495) [Of course, it also would've taken a lot longer because there is simply so many different ways those passwords](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=500) [could've been transformed, but then our success rate would've been much higher.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=505) [Mind you, if I had 2 or 4 or 8 or more GPUs that could've run the process and certainly that is something we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=509)[often see in dedicated password-cracking setups, then the speed obviously would've come down a lot longer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=515) [and if it could've run for say a day, our success rate would've been significantly higher.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=4&mode=live&start=521)

[Understanding salt and brute force attacks](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live)

[We can't talk about hashing without talking about salt and what salt does is helps us solve a very fundamental](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=0.5) [problem with the fact that a hash is a deterministic algorithm, and that problem is that if we can apply the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=6.5) [same algorithm to the same piece of input text and always get the same output or the same cipher,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=12.5) [once you start building up a database of hashes and their corresponding ciphertext for a particular algorithm,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=18.5) [it becomes very, very easy to crack the hash. And a really good example of this is using Google to crack a hash.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=25.5) [So let's jump over to Internet Explorer and I'll show you what I mean by that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=33.5) [Now let's imagine that a system has been breached and hashed passwords have been exposed and an attacker now](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=37.5) [has access to those. In many cases all it takes is just copying that hash, pasting it, searching for it, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=43.5) [here we have a result. So, first of all that hash appears to be cryptographically secure, certainly it's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=51.5) [obviously a very random output, but what's happened is when we've googled it, we've found a result.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=57.5) [So let's drill into this result and see what is behind it. Now what we see here is that that string that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=62.5) [just pasted in is a SHA-1 hash and it is a SHA-1 hash of P@ssw0rd. Now, of course, this is P@ssw0rd that has been](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=69.5) [obfuscated -- capital P, @ instead of an a, and a 0 instead of an o.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=77.5) [So someone might've thought they were being quite secure by creating a password structured like this,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=83.5) [now in reality that's just a very common mutation so really no matter how it's stored that isn't a very secure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=88.5) [password, but then of course because it was stored just as a SHA-1 hash, so exactly the same way that ABC](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=94.5)[stored their passwords, all it takes is for one other example of that to appear somewhere on the internet](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=100.5) [alongside the SHA-1 hash and a quick Google and there's the result and the majority of the time when a system](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=107.5) [is breached and the passwords have just been directly hashed, it only takes a Google of that hash to find](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=113.5) [the plain text version online; it's that simple. Google is an extremely effective hash-cracking tool.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=119.5) [Let's jump back over to the slide deck.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=126.5) [Now googling every single hash in a large dump, say 50,000 password dump like ABC, is not really practical and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=129.5) [this is where the role of rainbow tables comes in. So, rainbow tables were invented as a means of pre-computing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=136.5) [a large number of hashes so that it was very easy to compare them to a set of breached accounts.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=143.5) [So what a rainbow table does is it has all of these hashes and corresponding plain text passwords stored in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=149.5) [very, very large files and then the files are structured, such that a rainbow table tool such as RainbowCrack,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=155.5) [which we'll look at in a moment, is able to move through the tables and the breached hashes extremely quickly.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=162.5) [So it can be a very, very effective way of cracking passwords that are simply hashed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=168.5) [Now the problem with rainbow tables is that they can be very, very large.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=174.5) [Let's go and take a look at RainbowCrack and I'll give you an example of that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=179.5)[RainbowCrack is probably one of the most popular pieces of software for using rainbow tables against a set of hashes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=184.5) [Now RainbowCrack makes it possible to download a number of pre-computed rainbow tables and if we scroll down](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=191.5) [a little bit, we'll see that there are rainbow tables for NTLM, for MD5, and for algorithms such as SHA-1.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=196.5) [Now the thing about rainbow tables is that they're generated against a particular space of characters,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=204.5) [so, for example, ASCII characters for passwords ranging from 1-7 characters.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=209.5) [Now if you wanted to grab those, that's a 65-GB rainbow table. Take that up to 1-8 characters and suddenly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=215.5) [we're up to 576-GB. Keep going up and if you want mixed alphanumeric from 1-9 characters, suddenly you need](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=222.5) [864-GB to fit that rainbow table in. Now that doesn't include things like punctuation either.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=230.5) [So even though there is an absolutely massive key space or, in other words, the total number of passwords that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=238.5) [could be contained in this table, it's still going to miss a lot of possible passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=243.5) [The other thing is, is that downloading 864-GB is not a trivial task -- that's going to take a while.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=249.5) [You can generate them yourself, but generating rainbow tables is also a very, very computationally expensive](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=255.5) [process; it would take a very long time to generate 864-GB with the rainbow tables and, again, there's a lot](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=263.5) [of passwords that would not be catered for simply because their fallouts are that 1-9 character alphanumeric range.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=270.5) [So rainbow tables are rather unwieldy, but they can be extremely effective.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=277.5) [Although they are only effective, again, because of the deterministic nature of the hashing algorithm and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=283.5) [simply because we know that if we're just directly hashing a piece of plain text, we'll always get that same output.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=289.5) [So the rainbow table has gone and pre-computed that and that's what makes them work so well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=295.5) [Let's jump back to the slide deck.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=299.5) [To try and mitigate the risk of that predictability, so the risk of the fact that that hashing algorithm is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=302.5) [deterministic and will always produce the same result for the same input, we have the concept of a salt.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=308.5) [Now a salt is merely a sequence of random bytes that you add to the password before it's hashed and the idea](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=314.5) [here is that we're trying to break down the predictability of the hash.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=321.5) [What this means is, that if we take the same plain text password, but we add different sets of random bytes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=325.5) [to them, then the output, the ciphertext, is going to be different for each of them even though the original](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=332.5) [password is the same. Now the way this works in practice is that the resulting hashed password, the one that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=338.5) [was hashed with the salt, is stored in the database and you also store the salt in the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=344.5) [So what it means is, is that when the user comes to log on later on, we can pull out the salt for their username,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=349.5) [we can add it to the password that they just provided, the plain text one, we can hash them both together,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=356.5) [therefore repeating the process they went through at registration, and then compare it to the one in the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=362.5) [So there's sort of another step in there where we have to actually pull that salt out before we hash the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=368.5) [password, but what it means is, is that we now have the unpredictability.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=372.5) [It doesn't matter how many people use the same password. So long as you've got a nice, long, random salt,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=377.5) [you'll never end up with the same hash. What this means is that you're rainbow tables become useless because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=384.5) [you can't pre-compute the hash. You'd have to have a separate rainbow table for every single hash in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=390.5) [database simply because it has a different salt. So once we have that salt, the rainbow table is useless and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=395.5) [also googling the hash is useless because it only makes any sense with the salt and because the salts are random,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=403.5) [it completely destroys that predictability, therefore rendering both Google and rainbow tables useless.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=409.5) [Salts make passwords significantly more secure once they're hashed, there's no doubt about that,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=418.5) [but ultimately cracking that salted hash always comes down to a matter of time and earlier on, when we used](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=425.5) [RainbowCrack to break the hashes stored by the membership provider, we saw that ultimately we managed to crack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=432.5) [a significant portion of the salted hashed passwords anyway. So what this is about is how fast can we compute](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=438.5) [those hashes. Now the things about GPUs is that they are extremely effective at computing hashes and we just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=445.5) [saw how Hashcat can compute up to 7.5 billion hashes per second, so that was the NTLM hashes, and then it was](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=453.5) [slightly less for MD5 and slightly less for SHA-1, but we were still looking at over 2 billion hashes per second](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=460.5) [with SHA-1. Now when we talk about consumer hardware, it's consumer hardware like this.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=467.5) [So this is that AMD Radeon 7970 graphics card that Hashcat referred to on their website and that's the one](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=473.5) [that I have in my PC, which I used earlier on to break that membership provider.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=479.5) [So keep in mind that, first of all this is consumer-level hardware, it does have a cost of only several 100 dollars,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=483.5) [and furthermore, this is consumer-level hardware from early 2012.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=489.5) [So by the time you're watching this video, inevitably there are much faster, much more efficient pieces of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=494.5) [hardware out there and probably for less money too.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=500.5)[Now, of course, we also made the attack on the membership provider passwords very efficient by using a dictionary.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=504.5) [So we had a list of common passwords and often those lists are built up from previous breaches, particularly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=510.5) [like in the case of Sony where those passwords weren't protected at all, they were just plain text so that's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=516.5) [very, very easy to build up a database of typically user passwords, and that allowed us to very effectively](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=521.5) [crack those hashes, even though our through-put was "only" about 70,000 hashes per second.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=527.5) [So, as I mentioned earlier, we could've achieved a much higher through-put so we could've gotten into the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=534.5) [billion per second by using mutations, but of course you're trying much more random combinations of characters then.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=539.5) [The thing is though for an attacker, where they've got time, they have the ability to do that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=545.5) [So, once again, all of this just comes back to -- how do we make it hard enough to protect our passwords?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=550.5) [How can we increase the amount of time required to crack them?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=556.5) [So let's move on to looking at how we can do that by using different approaches to the way we hash our passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=5&mode=live&start=560.5)

[Slowing down hashes with the new Membership Provider](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live)

[When I cracked the hashes with Hashcat earlier on, and remember that was the salted hashes as well, so it wasn't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=0) [just direct hashing, those hashes have been generated by an ASP.NET web forms application created with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=6) [Visual Studio 2010 and that's what you see in front of you just here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=13) [Now the thing about that website is that by default, the membership provider that it uses is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=17) [SqlMembershipProvider, and the problem with the SqlMembershipProvider is it is just a SHA-1 hash with a salt.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=23) [Now many people have felt for a very long time that that is secure, but of course as we've just seen, it really](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=31) [is not a secure way of storing passwords and it's not secure simply because it is too fast.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=37) [So this is actually a very counter-intuitive concept to developers, where normally we want to make things as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=43) [fast as what we can, but in the case of hashes, we really want to make them as slow as what we can.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=48) [We need to decrease the rate at which hashes can be computed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=53) [Now fortunately things have improved a lot in later versions of ASP.NET and in particular with the templates](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=57) [that Visual Studio gives you. So what I want to do now is create a brand-new project and this time I am going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=64) [to create an ASP.NET MVC 4 application and it doesn't matter where it goes, let's just create that new application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=70) [Now the thing about an MVC 4 application created in Visual Studio 2012 is that you won't see a declaration](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=82) [for the membership provider. What's important though is what is built into the project and what I'd like to do](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=89) [is jump over and show you what that implementation actually looks like and because large parts of ASP.NET are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=96)[now open source, I could show you exactly what that looks like straight off the web.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=102) [The implementation within ASP.NET MVC 4 in a new template that you create from Visual Studio 2012 makes use](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=108) [of System.Web.Helpers and, in particular, the Crypto class. Now within the Crypto class, there is a very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=115)[important method, which is the HashPassword method and this is very kindly documented just above to explain](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=122) [exactly what that hashing algorithm does. And the first thing we can see here is that it uses PBKDF2 or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=129) [password-based key derivation function. Now what PBKDF2 lets us do is actually create multiple iterations of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=136) [the hash. So rather than just applying one SHA-1 hash, it is able to apply many SHA-1 hashes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=144) [Now obviously if you hash twice, it will take twice as long; if you hash 20 times, it will take 20 times as long.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=151) [What's very important in this implementation is that it hashes 1,000 times.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=158) [So, what that means is, is that the process of actually calculating a hash from a piece of plain text should](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=163) [take about 1,000 times longer for the passwords created in an ASP.NET MVC 4 application created with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=169) [Visual Studio 2012. So think about the example earlier on where it took about 14 minutes for us to crack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=177) [over 60% of the passwords from that ASP.NET membership provider created by Visual Studio 2010.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=184) [If we expand that out 1,000 times, that 14 minutes becomes 10 days.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=190) [Now that is a significant increase. There is a huge difference between being able to sit down for a quarter](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=196) [of an hour and crack a large number of passwords to it taking a week and a half, but it also reminds us that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=202) [none of this is about being 100% secure; this is just about making it harder, and the question then becomes](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=209) [how hard should we make it, is 10 days enough? Many people would argue that it's not and, in fact, it's worse](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=216) [than just 10 days because if we had 10 GPUs and certainly there are many examples of attackers having 10 or more](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=222) [GPUs, that 10 days is certainly going to be 1 day and, arguably, 1 day is just a little bit too easy to crack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=228) [a significant portion of the passwords. Unfortunately, with the implementation we have in ASP.NET today,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=236) [we can't customize the number of iterations - it is 1,000 full stop.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=242) [The other thing is, is that we are looking at an ASP.NET MVC application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=247) [Even in Visual Studio 2012, the ASP.NET web forms application is still not particularly secure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=252) [The hashing implementation does change, but the speed is still very similar to the original hashing implementation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=259) [that we saw in Visual Studio 2010. So a combination of that and the fact that 1,000 iterations may just not be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=265) [quite enough means that we do need other options. Whether or not that other option is required for your implementation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=271) [is going to be very specific on the type of application, the type of infrastructure it runs on, and who you're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=278) [trying to protect the data from. So let's have a look at some other options for stronger hashing in ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=6&mode=live&start=285)

[Other stronger hashing implementations](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live)

[So far in all the ASP.NET examples we've been looking at and also in the ABC example that we looked at earlier on,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=0) [the hashing algorithm used is SHA-1. Now another hashing algorithm that has become quite popular is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=6) [BCrypt hashing algorithm and fortunately over at NuGet there are a number of different options for implementing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=13) [BCrypt within an ASP.NET application. One of the ones that's very popular is BCrypt.Net and the thing about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=19)[BCrypt is it makes it very easy to define the workload that you want to apply to the hashing algorithm.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=25) [So by workload, we mean how much CPU overhead is required in order to create the hash.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=32) [BCrypt allows you to exponentially increase that workload so it's very easy to say, for this particular](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=37)[application, I want the hashing algorithm to work twice as hard as what I might normally like.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=44) [So in other words, what that means to an attacker is it's going to take twice as long to crack the hashed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=50) [passwords if they go through and recompute them using the BCrypt algorithm.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=54) [Now, of course, that also means that when someone registers on the site or when they log on or if they reset](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=59) [their password, so any circumstance where the plain text is going to need to be hashed, it means it's also](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=66) [going to take twice as long, but you've got to remember the durations that we're talking about here --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=71)[will an end user notice if the process takes, say, 100 ms instead of 50 ms? No, they won't.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=76) [However, will your infrastructure notice if you have 100 people trying to log on at the same time and the CPU](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=84) [overhead is doubling for every single one of them? Well yes, that may be a problem.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=90) [So this is one of the challenges of choosing how to implement hashing -- how do we strike a balance between](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=95) [security, usability, and impact on the infrastructure. Another NuGet package, which is quite popular for hashing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=101) [is Zetetic. Now one of the things about Zetetic, if we jump over into the project side, is we'll see that they](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=108) [actually support a few different ways of doing their hashing. So let's just jump over to the blog I have about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=115) [stronger password hashing. Now one of the neat things about Zetetic's implementation is that it can be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=121) [substituted for the existing SqlMembershipProvider so you can drop this into an ASP.NET application with a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=129) [little bit of reconfiguration. The other thing about Zetetic's implementation is that they support BCrypt,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=137) [which we just looked at in another NuGet package, and they also have an implementation that uses 5,000 computations](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=143) [of PBKDF2. So in other words, 5 times as many as what we get in the ASP.NET membership provider for an MVC 4](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=149) [application built in Visual Studio 2012. So now if we go back to those calculations before, the old 2010](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=158) [membership provider only took 14 minutes for us to break over 60% of the passwords; the new one would be about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=165) [10 days; if we used Zetetic's implementation, now we're talking about 50 days.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=171) [So just by carefully choosing the hashing algorithm you're going to use to store your passwords with, which is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=177) [an important decision, but also an easy one to implement, it can make the difference between breaking those](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=183)[hashes, even salt hashes, in only 14 minutes vs. 50 days and that is a very, very fundamental difference.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=189) [It has a huge impact on the success an attacker will have if they gain access to the system.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=198) [So this is why it is just so important to carefully choose that hashing algorithm because once you get it wrong,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=204) [it is too late to get it right once hashed passwords are out there in public.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=7&mode=live&start=210.5)

[Things to consider when choosing a hashing implementation](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live)

[Getting password hashing right can be a difficult process. For example, earlier on I mentioned that slowing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=0) [down the hashes is great in terms of security and in reality, users won't notice it because you're only talking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=7) [about 10s or, at the most, 100s of milliseconds anyway, but of course that is actually overhead.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=12) [The reason it's taking longer is because the CPU has to work a lot harder so that can cause an adverse reaction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=18) [on the server. One of the concerns many people have is that does that make it easier to create a denial of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=24) [service attack on the server. So what they mean by that is if an attacker started to try and cause automated](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=31) [logins knowing full well that a logon is a very computationally expensive process, could that actually slow](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=38)[the application down so much that it wasn't able to server legitimate requests?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=44) [Now there are mitigations against that. So, for example, limiting the number of logons that can happen from](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=49) [a certain IP address, but it does introduce new problems that you don't have with faster hashing algorithms.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=54)[The other thing is, is that if we find a good balance for an application today, is it still going to be a good](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=62) [balance tomorrow. So many of you are probably familiar with Moore's law, which is the law that says approximately](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=69) [every two years, computing power is going to double. Now if you think about the lifespan of a web application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=74) [and six years is probably a reasonable lifespan for many applications, are we prepared for the cracking process](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=80)[to be eight times faster towards the end of the lifespan of that application?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=87) [So think about that 10 day example again with the ASP.NET membership provider in Visual Studio 2012 and MVC 4 --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=91) [are we happy for that 10 days to come down to 1 and a bit days?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=98) [The other thing about this whole hash cracking discussion is that it's a little bit of an unfair competition](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=105) [and it's unfair insofar as when we create a web application, it's almost always the CPU which is doing the hashing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=110) [and the CPU is slow. Now, unfortunately, what makes it unfair is that when attackers want to crack hashes,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=119) [they use the GPU and the GPU is very fast. In fact, it's around about 100 times faster than the CPU.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=125) [So for us as developers, we're working with equipment, which is extremely slow in trying to find the right balance,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=133) [and then an attacker gets something like 100 fold advantage on us when they go to try and crack the hashes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=141) [There have been cause for applications to use GPUs when they're legitimately creating hashes, so at registration](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=146) [and logon, but unfortunately that's infrastructure, which really, really exists in server environments for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=152) [the purposes of creating hashes. Now that might be something that's feasible, particularly for very, very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=158) [large sites with huge audiences and also huge budgets, but for the majority of sites, the reality of it is -](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=164) [the creation of hashes is going to happen in the CPU. So, unfortunately, we're already going to be well behind](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=172) [the attackers in terms of speed. One way we could approach this is that over time, as Moore's law holds true](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=178) [and computing power increases, we could always change the hashing algorithm.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=187) [So maybe every couple of years we double the number of iterations of PBKDF2 or we increase the BCrypt workload.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=191) [The problem then, of course, is that you've still got to be able to log on against the hashes that are already](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=199) [stored in the database, so you have to be able to give people the ability to apply the old hashing algorithm](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=204) [to existing passwords, but the new hashing algorithm to new passwords.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=210) [Now you could always upgrade people as they come along, so login then uses the old hashing algorithm and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=214) [rehash the password with the new one and store it, but then you also need to track which hashing algorithm has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=219) [been used against which account. So there is a little bit of overhead there to build into the web application,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=225) [but certainly that's one way that this problem can be mitigated.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=230) [The other problem with that though, of course, is that it's only going to help protect the accounts that people](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=234) [have logged in with since the new hashing algorithm has been implemented.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=240) [You might still have a large number of accounts that have been implemented with the hashing algorithm,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=244) [which you now deem to be a little bit too fast. The bottom line with all this though is that there is simply](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=248) [not a "one size fits all" answer. Every situation is different. The amount of money people have to spend on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=257) [security is different, the number of users they have is different, particularly simultaneous users, which](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=264) [impacts the overhead on the CPU when hashing algorithms are slowed down, and of course the other thing that's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=270) [really different is the value of the asset they're trying to protect and not only that, but who they expect](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=275) [the attacker to be. Your approach to protecting a valuable asset that might have military value and there](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=282) [might be nation states that are looking to gain access to it is going to be very different to how you approach](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=288) [something like a web forum, where the reality of it is your most likely attacker is going to be a hacktivist](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=293) [with very limited funds. The point of all this is to have an objective discussion about this with your](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=299) [development team. Understand all the different factors that go into password security and hashing and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=306) [objectively come to a conclusion with a full understanding of what the risks of that decision are.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=8&mode=live&start=312)

[Understanding symmetric and asymmetric encryption](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live)

[So far in this module about insecure cryptographic storage, we've really only been talking about hashing and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=0) [password storage, but of course, many times you do need to cryptographically store data and be able to get it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=6) [back to the plain text version. So now we are actually talking about encryption and when we talk about encryption,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=12) [the first thing to understand is that there are two very different types of encryption -- symmetric encryption](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=18) [and asymmetric encryption. When we talk about symmetric encryption, we're talking about having a single](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=23) [private key, so one key that we keep secure and we do not expose to anybody else, and that key is used for both](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=29) [the encryption and the decryption. So an example of where we might use symmetric encryption where we only have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=37) [that one key is where all of the encryption and decryption happens internally within the application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=42) [So, for example, if we wanted to store credit cards, we might have someone enter it in plain text into a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=49) [web form, the application would be the one which then encrypts it, and at the right time, the application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=55) [would be the one that also decrypts it. You never need to give the ability for an external party to perform](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=60) [the encryption for you. When we talk about asymmetric encryption, that's when we're talking about both a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=67) [public key and a private key and you'll often see this referred to as public key encryption.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=73) [So, a really good example of where you will often see public key encryption yourself is with SSL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=79) [When you go to an HTTPS address on a website, it is your browser that needs to do the encryption before the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=84) [data is transmitted over the network. It's the server, which then needs to do the decryption.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=92) [So the server holds the private key; the server is the only one able to decrypt that traffic, but of course,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=97) [you had to encrypt the packets to begin with on your side using the public key that was served up when you](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=103) [originally loaded the website. And once again, just to reiterate, both of those mechanism of encryption do](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=110) [provide the facility to decrypt. So, again, it is a very fundamentally different approach to hashing,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=117) [which is not encryption and can't be unhashed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=124) [The big challenge with encryption and, indeed, the reason why we try and avoid it for password storage where](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=129) [we have another way of doing it is the challenge of key management.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=135) [The problem, of course, is that if the private key is disclosed then the encryption scheme is rendered useless.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=139) [So, keeping in mind that the idea of cryptography is to protect the data should it be exposed, if an attacker](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=146) [can also obtain that key, then obviously they can unlock the data and the whole scheme falls apart.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=153) [The problem with key management comes when you consider that an attacker often has access to, not only the data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=159) [when they breach a web application, but also the files on the web server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=167) [Now if, for example, the private key is stored in the Web.config and the attacker has managed to obtain both](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=171) [the data and the source files of the application, then you have a serious problem because the attacker has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=177) [the private key and, again, that's one of the major value propositions of hashing -- there is no key management](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=182) [so there is no problem of protecting the keys. If the attacker owns everything on the box, the web application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=188) [and the data, they still have to go through the hash cracking process in order to get access to the original](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=196) [plain text. So in many ways, encryption is really undesirable; it's not something we want to do unless we really,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=201) [really have to do it because once we start encrypting, then we have to deal with the challenge of key management.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=9&mode=live&start=208)

[Demo: Symmetric encryption using DPAPI](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live)

[One of the best ways we have of dealing with symmetric encryption in ASP.NET is to use the Data Protection API](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=0) [or DPAPI. Now what makes DPAPI so useful is that you don't have to worry about the key management.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=8) [At least you don't have to worry about it within the application because it uses the machine key on the server](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=16) [that you're running on. Now this is exactly the same mechanism that we used earlier on in the Security](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=22) [Misconfiguration module when we encrypted sections of the Web.config.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=27) [Now as a result, you also have the same challenges as we discussed earlier on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=33) [So challenges such as when you might have a web farm with multiple web frontends and possibly different machine](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=37) [keys, their issues we've already discussed and they're the same issues that apply here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=42) [Let's look though at how we can implement DPAPI easily within a web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=47) [So I'm going to take the existing one that we cracked a little bit earlier on and I'm just going to add a new](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=53) [web page here and in this case, we can just add a web form without any master page and we will call it Encryption.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=59) [Now what I want to do here is I'm going to add three labels and the first label I want to add will be our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=68)[‘Plain text' and we will just give it an asp:Label and we'll call that PlainTextLabel.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=74) [Now I'm going to want to create two more labels and the first one will be ‘Encrypted text' and we will make that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=83) [an EncryptedTextLabel and then we will also have some ‘Decrypted text' and that can be a DecryptedTextLabel.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=91) [And the purpose of these is probably pretty self-explanatory so let's now jump into the source code and implement it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=104) [And what I'm going to do here is just start out by defining a string that we will call our secret and in this case,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=109) [we can call it "My secret text." So before we can go and encrypt this, we need to turn this into a byte array.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=115) [So let's call this secretBytes and we will just jump into encoding, which is in System.Text.Encoding,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=123) [Unicode.GetBytes and we will pass the secret into that. Now what we want to do is we want to encrypt those](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=132) [bytes, so we will make a variable called encryptedBytes and we're going to call ProtectedData and ProtectedData](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=140) [happens to be in System.Security and then we're going to call Protect.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=148) [Now when we call Protect, we need to give it first of all the data that we want to protect, which will be our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=153) [secretBytes. Then there is an optional entropy value, which we won't use just for the moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=157) [And finally, we will need to give it a DataProtectionScope. So let's give it a secretBytes and then we don't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=162) [need any optional entropy and finally we need a DataProtectionScope and we'll give it the scope of the local machine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=169)[So, what we finally got here is a set of encryptedBytes. Now let's go through and do the decryption and, in this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=176) [case, our decryptedBytes because the first thing we're going to do is take those encryptedBytes and decrypt them.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=182) [It is also going to be the ProtectedData namespace and we are going to call Unprotect and we'll just go through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=189) [a very similar process again. So, in this case, we're going to give it the encryptedBytes, we're not going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=195)[specify any optional entropy, and the scope we're going to use is local machine, but of course, ultimately we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=200) [do want to get this back to a string so let's call this one decryptedSecret and we will just call enter Encoding](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=206) [again, so Encoding.Unicode.GetString and we'll give it our decryptedBytes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=213) [Now because we want to see what all that looks like, let's populate this into the labels.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=220) [So let's start with our PlainTextLabel and we'll set the Text value of that to our secret and then we'll look](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=224) [at our encrypted label and we'll set the Text value of that to our unencoded encryptedBytes because obviously](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=230)[we don't want to show encoded cryptedBytes. So we'll get the string again of our encryptedBytes and then,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=238) [of course, our decrypted data, which we will set as decryptedSecret, and let's run that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=247) [So here we go, let's step through this. So, first of all, ‘My secret text,' which was the original text that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=255)[started with, and ‘Encrypted text'. So obviously we are getting some non-ASCII character bytes here, which is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=260) [fine because ultimately this is an encrypted byte array -- we don't expect it to make any sense to us whatsoever.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=267) [The important thing is though, is that once we have decrypted, ‘My secret text' comes back out.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=274) [So what we've been able to do here is take the plain text, encrypt, decrypt, and we've never had to worry about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=279) [a key, which is great as we don't have to worry about the problems of key management.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=287) [Now what this means is, is that even if the web application in the Web.config get breached, so long as the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=291) [machine key set on the server remains secure and, of course, there are times when it won't, if the entire box](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=297)[gets owned by an attacker, we're going to have a problem, but if the entire web application and the database](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=303) [get owned, our encrypted data is going to remain safe. And that's what encryption is all about.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=309) [Now let's just go back and look at one setting that we ran through quite quickly and that is the Scope.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=314) [So, we've used the LocalMachine scope here. There are actually two different options for our DataProtectionScope,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=320) [obviously we've seen LocalMachine, the other one is CurrentUser, and the difference is quite important.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=326) [Now both of these scopes have discrete purposes and they're both relevant in different circumstances.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=331) [So, the thing about the LocalMachine is that any process running on that machine can then decrypt the data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=336) [that is encrypted with the scope of LocalMachine. So, if you have multiple applications on the machine that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=343) [may actually run under different identities, LocalMachine makes sense as they all have access to the data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=349) [Alternatively, if you wanted to prohibit another user from decrypting that data who had access to the machine,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=355) [then changing the Scope to the CurrentUser will make sure that they can't get access.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=362) [So there is a very small, but very important semantic to the way we choose that Scope.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=366) [Setting up your application using either Scope will work, the difference is who you're trying to protect the data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=372) [from and what you're trying to give access to the encrypted data, so think carefully before choosing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=378) [CurrentUser or LocalMachine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=10&mode=live&start=384)

[What's not cryptographic](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live)

[The last thing that I'd like to look at in this module is some myths and misunderstandings about cryptographic storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=0) [I've seen many cases in the past where someone believes that they're implementing some cryptographic protection,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=8) [whereas in reality, it's doing just about nothing to protect the data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=14)[Now often what we're talking about here is just some simple string transformation processes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=18) [So, for example, character rotation. Now a good example of character rotation is what we refer to as ROT13.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=23) [So, the way ROT13 works is we take a range of characters, so let's say characters from the alphabet in uppercase](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=31) [and also characters from the alphabet in lowercase and what happens is we rotate them and when we're doing the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=38) [rotating, we're rotating them by 13 positions. So, A is the first letter of the alphabet and N is the 14th letter](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=44) [of the alphabet -- it's 13 positions further along, so a capital A goes to capital N, lowercase a goes to lowercase n.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=52) [Now this might look cryptographic to some people, but it's just character substitution and clearly it's very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=61) [easy to take the output of a ROT13 process and convert it back to the original text.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=68) [So there's really no cryptographic benefit whatsoever in character rotation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=73) [Another good example is encoding, so often seen Base64 encoding used as a method of "encryption."](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=80) [Now clearly, Base64 encoding is not cryptography. It's really just a means of taking a byte array and converting](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=89) [it into a text encoding scheme; it is reversible -- Base64 encode, Base64 decode - there is no key involved.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=97) [Now having said that, we do often see Base64 encoding used as a method of storing bytes in cryptography.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=106) [So when we saw those hashed ASP.NET membership provider passwords and the salts in the database right back at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=114) [the beginning of this module where we breached the hashed passwords, all of those were Base64 encoded before](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=121) [they were stored in the database, but that was really only to get byte arrays into ASCII characters so that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=127) [could store them in VARCHAR fields. So that wasn't a means of protecting data, it was a means of storing data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=133) [that was already protected. If you're not sure about whether a particular process provides any cryptographic](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=140) [protection or not, a really good reference is Kerckhoff's principle, and what Kerckhoff said is that --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=146) ["One ought design systems under the assumption that the enemy will immediately gain full familiarity with them."](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=151) [Which means ask yourself this -- am I happy to publicly disclose the mechanism with which I have transformed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=158) [the data I want to protect? Now that doesn't mean disclosing the key, assuming a key exists, and obviously it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=166) [doesn't if we're only doing character rotation or character encoding, but if I disclose that mechanism,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=173) [can an attacker then take that data, which I believe that I've protected, and get it back to plain text?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=179) [If they can, then it's really not a means of cryptographic storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=186) [And just as one final example of the misunderstandings around cryptographic storage, I wrote a post in early 2013](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=192) [which drew some examples out of a question on stack overflow. The question has since been deleted, but originally](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=199) [what it said is, someone has a table where they want to store some login data and they want to encrypt the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=205) [password and I found it very interesting to see some of the answers, which is obviously the way other people](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=211) [are protecting their passwords. So, one of the first answers effectively just Base64 encoded the plain text,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=217) [which as we just saw is really useless. One of the next answers simply did character rotation.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=222) [It may not have been precisely ROT13, but it was a simple character rotation, which again provided 0 value.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=229) [And then there was another example around encoding and, of course, again it provides 0 value.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=236) [So the point I'm trying to make here is that cryptography is very, very commonly misunderstood and in many cases,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=242) [there are people out there building these systems, using encoding or using character rotation, believing that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=249) [they are implementing cryptography when they're simply not.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=11&mode=live&start=256)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live)

[Let's summarize the module, and the first thing that I want to reiterate is that cryptographic storage really is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=0) [the last line of defense. So, cryptographic storage is what's going to save us after a system has already](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=7) [been breached. It may have been breached by a SQL injection flaw, it may be that someone has managed to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=13) [remote into a machine, it may even be that backups have been disclosed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=20) [There are many different ways that a system can be breached and cryptography is the last thing that's going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=24) [save us when that happens and that's why it's so important to get that right.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=30) [Of course, it's very important to get everything upstream right, but with security it's all about layers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=34) [So how can we try and protect against the risk of SQL injection, but when that fails, how can we protect the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=41) [data, which might've been exposed? We spoke a lot about password hashing and, again, the thing with password](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=47) [hashing is trying to slow the process down and increase the randomness.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=54) [When we looked at that, we saw that even the most secure hashing mechanisms were never about getting us to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=60) [100% secure, it was all about making it much, much harder. So how could we get the time taken to crack those](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=66) [passwords from 14 minutes up to 10 days up to possibly 50 days? That presents a very, very different value](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=73) [proposition to an attacker. Some of the mechanisms we looked at to do that were obviously things like PBKDF2,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=80) [which is built into an ASP.NET MVC 4 application provisioned in Visual Studio 2012 or libraries such as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=87) [BCrypt.Net or Zetetics, both of which are available for free on NuGet.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=95) [And of course, even though salt didn't save us in the attack that we used against the ASP.NET membership provider](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=100) [from Visual Studio 2010, it's still extremely important. We need to have that randomness, but salt alone is not](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=106) [enough and that's why we correct those salted SHA-1 passwords so quickly.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=114) [Encryption -- both symmetric and asymmetric encryption are important, even if we see it far less frequently than](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=119) [password storage, but again the problem with encryption remains decryption, it remains the key management.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=127) [Key management is hard and there are systems that are sold for huge amounts of money to try and abstract away](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=134) [the problem of key management. The Data Protection API does make it much easier to solve this problem,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=141) [but of course, it also introduces other problems. You do still have a key on the machine and you are going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=148) [have problems if you multiple frontends; you need to approach it differently then.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=154) [So that's where we need to consider the difference between having a scope that is the LocalMachine vs. a scope](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=158)[that is the user. And finally, those myths and misunderstandings about cryptographic storage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=164) [Character rotation is in cryptographic, encoding is in cryptographic, and as we saw, there are many people](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=170) [out there who think it is. Unfortunately, they just don't understand the difference between encoding or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=176) [character rotation and genuine encryption. So always refer back to Kerckhoff's principle: are you happy](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=182) [to disclose to the enemy the implementation of your algorithm, because if you're not, it means that you're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=189) [probably not storing the data securely.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m7-crypto-storage&clip=12&mode=live&start=195)

[Failure to Restrict URL Access](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to Part 8 of the OWASP Top 10 on Failure to Restrict URL Access.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=0) [In this module, we'll start out as usual by taking a look at how OWASP views this particular risk and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=9.5) [also as usual we will move on to performing an attack against a vulnerable application, which fails to protect](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=15) [URL access. We'll then move on to understanding some of the access controls we have built into ASP.NET,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=23) [because there are a few different ways we can apply authorization rules.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=29) [We'll then move on to talking about why role-based authorization is important and we'll take a good look at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=34) [the role provider within ASP.NET. And finally, we'll take a look at some other access control risks and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=40) [misconceptions because there are a few strange ideas out there about what constitutes an access control.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=46) [So let's move on and take a look at the risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=0&mode=live&start=53)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live)

[Moving on to the OWASP overview and risk rating, as usual, the Threat Agent can be pretty much anyone who can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=0) [access the application. Of course, when we're talking about failure to restrict URL access, we're really](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=6.5) [starting to talk about people who are able to access content that they shouldn't.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=12) [So, for example, could an anonymous user access a part of the website that is meant to be private?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=17) [The Exploitability is extremely easy because really what we're talking about here, very often just boils down](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=25) [to a user changing the URL and being able to access content, which they should not be able to.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=31) [So clearly, the theme that's going to keep coming up in this particular module is lack of access controls](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=37) [on the resource. The Prevalence is classified as uncommon and the Detectability average, primarily because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=44) [this is just such a simple, simple, simple flaw and fortunately, we don't normally see it slip into our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=52) [applications, although it still does happen to the extent that it warrants inclusion in the Top 10.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=59) [Part of what makes this risk a little bit tricky is that often access controls are implemented in a variety](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=66) [of different ways. So the overview here talks about configuration, it also talks about proper code checks,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=72) [and as we'll see as we move into the other parts of this module, there are numerous different ways we can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=78) [secure an application and we can do so at various levels within the application -- so making sure those levels](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=83) [are properly addressed is something that developers simply sometimes forget to correctly do.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=90) [The Technical Impact is moderate and, of course, it will depend on the nature of the application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=98) [If you consider the potential impact though of, say, an administrative facility not being properly protected](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=103) [and being accessible to an unauthorized user, certainly that technical impact could end up being quite severe.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=110) [As usual, the Business Impact is pretty generic. IT's really going to depend on the nature of the business](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=120) [and, of course, on the nature of the information that the website is protecting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=1&mode=live&start=125)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live)

[Let's look at a very simple, although not entirely uncommon example of failure to restrict URL access.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=0) [In front of us here is a very generic ASP.NET MVC application, obviously just uses the standard template, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=8) [you'll note that all the links on the application are again the standard set of links that you get in a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=15) [new project template. However, once we log in, we're going to see something a little bit different.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=21) [So let's proceed to the Login link and I'll log in with my User name and my Password and what we will now see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=25) [is that there is a new link up in the top right-hand side of the screen; there is an Admin link.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=38) [So obviously I am only now seeing this Admin link once I am logged in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=44) [If I proceed to the Admin page, I will see some Admin facilities. Now, of course, normally we would have features](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=49) [that might be designed to say "Edit content on the site" or maybe "Control permissions"; use your imagination.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=56) [It will obviously depend on the nature of the website. The important thing is, is that once I have logged myself on,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=62) [I now have access to the Admin page. We can see by the URL that the Admin page is just simply located in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=69) [/Admin path. Let's now log myself off and see what happens if we go back to that /Admin path.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=77) [So I'm definitely now not logged in. If we refresh the page, I should still see that there is no ‘Welcome Troy Hunt'](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=86)[message there. Now what I want to do is just jump up to the address bar and we will put /Admin again and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=92) [here we go -- we can see that I'm back on the Admin page. And just to make sure it's not cached, if we refresh](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=99) [that page, I am still very clearly able to view the Admin page even though I'm definitely not logged in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=104) [I've still got a Login link up on the top right-hand corner of the page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=113) [Now, of course, it's very easy to see what's happening here. The Admin link is only appearing when I am logged in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=117) [So obviously there is some form of security-trimming implemented to ensure that I only get links that are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=124) [relevant to me. Unfortunately, what's happened with this application is that there is then no access control](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=130) [on the actual Admin page itself. So this is a case where we have insufficient access controls.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=137) [Obviously the developer has gone to some lengths in order to protect the Admin page, but simply hiding a link](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=144) [really isn't an access control and, clearly, as we can see here, if this practice was implemented on a live](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=151) [website and we do often see things like this happen, there would be a serious security risk with the web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=2&mode=live&start=157)

[Risk in practice: Apple AT&T leak](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live)

[Let's take a look at the risk of failure to restrict URL access in practice.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=0) [So, in this case, we have a precedent and this precedent was from the middle of 2010 and it was frequently](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=4) [reported as being "Apple's Worst Security Breach." An actual fact, it was more to do with the way the AT&T](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=12) [Telco Provider made information about subscribers who had bought iPads available to the public.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=19) [Now the scale of this breach was quite impressive because what ended up happening was the details of 114,000](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=25) [iPad owners were exposed simply because a URL returned their personal information based on a simple HTTP request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=32) [As we move through the article, we'll see that the actual information that was exposed did only end up being](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=42) [the subscriber's email address, as well as the associated ID used to authenticate them on AT&T's network,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=49) [which is known as an ICC-ID. Now really it's the email address, which is the risk here, because what's happening](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=56) [is that this insecure service is leaking information about subscribers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=64) [That information may then be used in a fishing attack because attackers do know that the owner has an AT&T account](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=70) [and they know the owner has an iPad as well. So for an attacker, they were able to start building a profile](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=76) [about what their victims interests and relationships might be. What's interesting about this breach was that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=82) [it was really extremely simple in execution and, in fact, we can see here that all the attackers needed to do](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=88) [was provide that ICC-ID as part of an HTTP request. Clearly this was provided to a web service, which was](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=96) [intended to be consumed from the iPad itself, but when the request was formed correctly, the web service would](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=103) [happily respond with an AJAX response, including the email address of the registered user.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=110) [So really, all an attacker needed to do was enumerate through possible ICC-IDs and look at the responses they received.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=116) [So actually this is very similar to the insecure direct object reference risk, which we saw back in Part 4 of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=125) [the Top 10. So certainly Part 8 of the Top 10 on Failure to Restrict URL Access is very, very similar.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=132) [As we scroll through some of the other details of the breach, one of the things we see that often happens in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=139) [a breach such as this is the leaking of high profile identities.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=145) [So particularly things like military identities and we can see here that there are some very high profile people](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=150) [who have had their email address leaked and of course the fact that they're an AT&T customer and they own an iPad.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=156) [It's always a similar sort of story with celebrities or other high profile people -- attackers like to leak this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=163) [sort of information to prove a point. Even though this particular attack really didn't expose anything too](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=168) [sensitive, it wasn't peoples' home addresses, certainly wasn't credit cards, it didn't allow them to manipulate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=176) [any data, just the simple fact of exposing these email addresses did make this a very high profile attack and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=182) [certainly the legal fallout from this event went on for some years to come afterwards.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=190) [So with an understanding of what ‘Failure to Restrict URL Access' now looks like, let's move on to properly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=196) [protecting a vulnerable application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=3&mode=live&start=202)

[Demo: Access controls in ASP.NET part 1: web.config locations](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live)

[When we talk about access controls or authorization in any technology stack, there are always different ways](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=0) [we can go about the implementation of this, and I'd like to take a look at a few ways of doing this within ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=7) [So one of the easiest ways that we can put some restrictions around the URL in ASP.NET is to simply use the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=14) [location element in the web.config. It's very easy to say, take this path and only allow access via these users](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=21) [or these roles, it's an extremely simple configuration, and the other neat thing about doing it in the web.config](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=29) [is it does consolidate all those access controls into one nice centralized place.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=36) [We'll take a look at just what that looks like in just a moment. The other thing we can do is we can take any](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=42) [method in any class and apply what's referred to as principal permissions.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=47) [So right down at that very granular level, we can say these are the identities or these are the roles that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=52) [want to grant access to this method. So that is a very, very fine grain permission and, of course, it sits](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=60) [with the actual code that you're wanting to protect. One of the advantages about that approach is it means](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=66) [that you can define that access control in one place and then regardless of the path or the page that is actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=72) [calling that method, the protection will be the same across all those pages.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=79) [Moving on to MVC, it's very easy to protect an entire controller using the authorize attribute and we'll have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=84) [a look at that in just a moment as well. We can also drill that down a little bit deeper and protect individual](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=91) [controller actions using the authorize attribute and in both those cases, we can be quite specific about the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=97)[individuals or the roles that we want to grant access to either the controller or the controller action.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=104) [All of those examples are native features specifically designed around the process of authorization, but of course](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=111) [in addition to that at any point in the code, it's very, very easy to reference an identity or reference a role](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=118) [and build conditional logic about what should happen if somebody either is or is not in that role.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=125) [So we can always drop down to that very programmatic approach without necessarily using the native semantics of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=130) [the framework. Let's move on and have a look at how some of these look in practice.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=137) [This is the web application we looked at earlier on where that Admin path was accessible even when you weren't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=144) [logged in. So what we saw then was that really it was just the link to the Admin page, which was being turned](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=150) [off when the user wasn't authenticated, and obviously what we need to do is secure the Admin page itself.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=157) [So what I want to look at now is using the location element in the Web.config to lock this application down](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=164) [so that that path can only be accessed by authorized users. So the first thing I want to do is just jump right](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=171) [down to the bottom of our Web.config and then before we close off configuration, we're going to start with a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=178) [location element. Now for this to work, we need to define the path of that location element and in this case,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=184) [we know that that path is called Admin. So, under the location path, we can just go and implement the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=191)[system.web element again and, of course, this is very similar to the system.web that is already backed into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=197) [the default Web.config of a new ASP.NET project. The only difference is, is that this is only going to apply](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=204) [to that Admin location path. So, once we have the system.web, now we can start looking at authorization and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=211) [in this case, we're going to do two things within our authorization node and the first thing is, is that we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=218) [want to put in a deny declaration and what we're going to do is deny all the users who are unknown.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=226) [So the question mark syntax says, we do not know who the user is. So that's quite obvious -- we don't want to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=234) [let anyone in if they're not authorized, but of course, what this could mean is that if you were logged on,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=240) [so if the application could identify you, you weren't anonymous, it would mean you have access to the Admin path.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=247) [Now based on the name of that, it's probably pretty fair to assume that we really don't want any authenticated](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=254) [user being able to access the Admin path, we only want a subset being able to access the Admin path.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=260) [So what I've done is within the application I've created a role called Admin and we'll look at this in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=267) [next section when we talk about the role provider, but the way I can use that here is I can now put in an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=272) [allow declaration and I can define roles and I can define the Admin role.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=279) [This doesn't have to match the path, it just makes logical sense that there is a role called Admin and there is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=287) [a path called Admin. We could also comment or limit these roles, so we can put in multiple different roles,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=293) [but for the moment we're just going to work with the Admin. Let's save this and then we'll jump back over](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=299) [into Internet Explorer and see how this changes the experience that we had before.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=304) [So here we are back in Internet Explorer and I've left it where we finished up earlier on after testing that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=312) [we could actually load the Admin page whilst we weren't actually logged in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=318) [Now that we've properly secured this path, let's just give it a refresh and see what happens.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=322) [Okay, so this is quite important. What we're now seeing is the Login page and, in fact, what we'll see is not](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=328) [just the Login page, but in the URL we can see there is a return URL, which is the Admin path.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=334) [Now this is all native behavior of ASP.NET. If we try to access a resource that is protected via the location](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=341) [element and we don't have access to it, we're going to be asked to log in and after logging in, the application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=349) [is going to be able to take this return URL path and send us back to the resource we were trying to access.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=356) [So it's a nice little usability feature built right into ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=361) [Let's log in and make sure that I can actually still get access to that resource.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=366) [So, User name, Password, Log in, and here is the Admin page. Now I'm seeing this because I am a member of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=371) [Admin role and, again, we'll look at how that is implemented in the next section of this module.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=380) [Now whilst this works reasonably well, there is a little bit of a problem with it, particularly in ASP.NET MVC.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=386) [Let's jump back to Visual Studio and I'll show you what that is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=4&mode=live&start=393)

[Demo: Access controls in ASP.NET part 2: The authorize attribute](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live)

[So back in Visual Studio, this is a very configuration-centric way of declaring our access rights.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=0) [One of the things that is quite nice about this approach is that we could have many different location elements.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=6.5) [So clearly we've restricted the Admin path to the Admin role; we might restrict other paths just towards](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=12.5) [authenticated users; we might then restrict even more paths to other roles that we have in the system; and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=18.5) [all of that would be declared here in the one location in the Web.config.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=24.5) [Now the problem is, is that what we're really protecting here is just the path.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=29.5) [So this is the address that we see in the web browser. It's not really protecting the underlying resource,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=34.5) [which for an ASP.NET MVC application is the controller. The problem comes from the simple fact that as we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=42.5) [add routes into the system, we may well end up with routes that map through to the same underlying resource,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=49.5) [through to the same controller action, that they may be on a different path.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=57.5) [Let's go and implement that so that we can see the risk and then we'll change the way we implement our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=62.5) [authorization in order to properly protect it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=67.5) [So, we'll jump into the Solution Explorer and in App\_Start, we will have our RouteConfig.cs file and this is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=71.5) [just a very default route configuration file. Let's now go and add another route and we will see where the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=79.5) [problem is. So let's do routes.MapRoute and I'm going to name this route Admin and I'm going to give this route](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=86.5) [a URL of Administrator. Now there are various reasons why we might want to add additional routes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=96.5) [So sometimes we might want to add them for Search Engine optimization purposes; other times it's because an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=104.5) [application has migrated from another framework and even though the new structure is different, we want to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=110.5) [maintain the paths of the old application so that external links don't break.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=115.5) [There are many different, very, very valid reasons why we might have different routes that actually map through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=120.5) [to the same resource. So let's now actually map that through to the resource and we are going to create a new](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=126.5) [anonymous type and we will call the controller Admin and, of course, this is a controller that we already have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=132.5) [in place, and in this case the action is simply the index. So let's give that a rebuild and we will go back](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=141.5) [and check what this looks like in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=150.5) [Now, back in Internet Explorer, I want to check a couple of different options.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=155.5) [So the first thing I'm going to do is log us off, so we are no longer authenticated as an Admin, and the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=159.5) [first thing we'll do is try accessing the Admin path, which of course should now be properly protected because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=166.5) [we put the location element into the Web.config and that's great, we can't go any further, it's asked us to log in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=171.5) [Now let's try accessing the Administrator path, which is the one we just added with the custom route (Typing)](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=178.5) [and here you see the problem. We've been able to load exactly the same page that the Admin path was trying to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=186.5) [load, it's just that now we've done it through the Administrator path simply because there's another route.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=192.5) [Same controller, same controller action, same content on the page, it's the same view, but what we've actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=199.5) [done is circumvented the access control and this is precisely why that configuration-centric approach in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=205.5) [Web.config where we're protecting paths is not a very good paradigm for ASP.NET MVC.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=212.5) [Let's go back into Visual Studio and reimplement this a slightly different way.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=220.5) [Here's where we just left off. So we just added in that Administrator route.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=225.5) [I'm going to jump back over to the Web.config and entirely remove that location element because really it's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=229.5) [not the way we want to secure our MVC application. What we want to do is we want to protect the actual resource,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=236.5)[which in this case is really the controller. So let's jump into our controllers, into our Admin controller,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=242.5) [and this is the guy that we want to protect. Now earlier on in the slide deck, I mentioned that we can use](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=250.5) [the authorized annotation and I also mentioned that we can use it on the controller or on the action.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=255.5) [So one of the decisions we need to make here is -- are we trying to protect every single action on this](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=261.5) [controller with the same access controls or are we only trying to protect the index controller action?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=267.5) [The reality of it is -- this is the Admin controller. Anything that we put in here is really going to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=274.5) [related to an Admin feature. So in this case, what I'd like to do is actually protect the entire controller](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=281.5) [and what I'm going to do is simply add an annotation called Authorize.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=287.5) [Now if we just left the annotation like this, you wouldn't be able to access this resource unless you were](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=293.5) [authenticated; you wouldn't have to be a member of the Admin role.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=299.5) [So let's add a little bit more information to make sure that it's only Admins that are going to be able to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=304.5) [access this resource. So what we're going to do is we're going to say Roles = "Admin" and, of course,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=308.5) [it's similar to when we defined roles in the Web.config, we could comment or limit and we could have multiple](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=316.5) [roles here. In this case, though, all we want is the Admin. So let's rebuild this application, jump back to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=321.5) [Internet Explorer, and we'll see how it behaves now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=328.5) [Okay, so we are back in Internet Explorer and we're back where we left off before, where we are not logged in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=332.5) [and we are able to see the Administrator path. Let's give it a refresh and see what happens now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=338.5) [Okay, good news. It has actually asked us to log in. So it sent us back to the Login page, we've still got](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=345.5) [that return URL query string parameter in there, which points through to an administrator.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=350.5) [Let's now try just changing that to the Admin route and, again, remember that both of these routes actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=355.5) [map through to the same controller action and it's exactly the same result.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=362.5) [Back to the Login page, back to our return URL, and back to Admin path in that return URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=366.5) [So even though they are two totally separate paths, the access control is consistent and it's defined in one place](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=373.5) [because we've got that Authorize attribute on the controller itself.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=380.5) [If we wanted to do this just purely within the Web.config, we would have to keep adding location elements.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=384.5)[So we'd have a location element with a path for Admin and then we'd have another location element with a path](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=390.5) [for Administrator. Again, that is a very sort of configuration-centric model and it actually moves the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=395.5) [definition of that authorization away from the resource itself, the controller, and it puts it somewhere else,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=402.5) [the Web.config. That works pretty well for a web forms application because you don't generally tend to have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=409.5) [custom routes in web forms. There are certainly ways of doing it, but it's not baked in from the ground up](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=415.5) [like it is with an ASP.NET MVC application. So certainly this is still a viable option for your web forms](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=421.5) [applications, it's just not the way you really want to secure those ASP.NET MVC applications.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=5&mode=live&start=428.5)

[Demo: Role based authorisation with the ASP.NET Role Provider](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live)

[On a number of occasions throughout the modules in this course, I've made reference to the ASP.NET](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=0) [membership provider and, indeed, we've implemented it in a couple of places, we've even broken it before when](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=5) [we looked at cryptographic storage. The thing about the membership provider is that it is an extremely easy](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=10) [to use implementation if you want to get up and running quickly with facilities such as Registration and Login.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=16) [And for the application that we saw earlier on, the only code that I'd implemented before we started using it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=23) [and exploiting it was simply the Admin page, which was only a controller and a view, and the link through to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=28) [the Admin page, which just displayed if the current user was authenticated.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=34) [What you see in front of you here is the database structure that is automatically created by a brand-new](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=38) [ASP.NET web application when you try and access a feature that uses the membership provider.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=44) [So long as you have a valid connection string in the application and the user in that connection string has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=50) [the rights to create objects, which incidentally is something that should only be the case while the application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=56) [is set up, after which you should apply the principal of least privilege, and it would only have data reader](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=61) [and data writer access, this is the structure that it creates.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=65) [I want to talk a little bit about the role provider now and we're going to focus on these three tables to the left here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=70) [We're going to focus on Roles, we're going to focus on UserProfile, and we're going to focus on the UsersInRoles.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=76) [Let's take a quick look at what is in these tables and then we'll go and add a new user and we'll also give](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=85) [that user administrative privileges, just to test how that role provider really works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=90) [So I'll just jump over to a query window and the first thing I want to look at is, let's take a look at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=95) [everything that is in that UserProfile table and as it happens, there's just one record -- it's the one that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=100) [I created for myself, and I've got UserId 1. Let's now take a look at what is in the role table.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=106) [Again, very simple -- only two columns, it's either in #1, and there is a role called Admin.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=114) [It's probably now pretty obvious what we're going to find in the Users in the Roles Table and there we go --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=122) [UserId #1 is in Role #1. So that's just a very, very simple map between our UsersInRoles and that's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=128) [all that's required to grant people access to a role, of course, once they have a record in the database.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=135) [To demonstrate this, let's jump back over to Internet Explorer, we'll create a new user, and then we'll come](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=141) [back here and we'll give them admin rights.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=146) [Okay, back in Internet Explorer. Now let's go and register a new account and this account can be johnsmith](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=149) [and we'll give john a Password (Typing) and now let's register.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=159) [Great, so john is logged on. We don't have a link to Admin though and if we try and access the Admin path,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=167) [which is now properly secured via the authorized attribute on the controller, we also don't have access.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=173) [Let's go and give john some Admin access so it's back to SQL Server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=178) [Back in SQL Server, let's take a look at our UserProfiles and, of course, we now have a second record in there](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=184)[for johnsmith. Now all we need to do to give john access is we just need to insert a new record into our](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=189) [UsersInRoles and in this case, our UserId is going to be #2, which is john's ID and of course the RoleId is #1,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=196) [which is the Admin. Now once we execute this, we should immediately have john in that role.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=205) [Let's flip back to Internet Explorer and see if that works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=212) [Back in Internet Explorer, every time a request is made for a resource, which uses the role provider,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=217) [it will go back to the database and check the membership.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=223) [So as soon as we click Home, here we go. We've now got an Admin link because it's gone and found john in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=225) [that mapping table and, of course, we can now access the Admin page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=232) [So that's a really, really easy way of doing your role management, but of course it does require access to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=237) [the database and that's really not the way we want to stand up an application where this is going to be a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=243) [continual process, particularly when you've got end users involved.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=249) [Let's take a look at how we can implement this very, very easily directly within Visual Studio.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=253) [One of the great things about the membership provider and about the role provider is that there are a lot of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=260) [great hooks built right into the framework that make it very, very easy to work with.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=265) [So as an example, if we wanted to build a facility for an administrator to manage role memberships,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=270) [so rather than going to the database and writing SQL statements, they actually wanted to have an interface where](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=276) [they chose somebody's name and then they chose a role and clicked the Submit button and it all happened automatically.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=281) [It's very, very easy to do with the role provider. Now we won't implement the whole thing, but just by example,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=287) [all we need to do is access Roles and we'll need to reference System.Web.Security.Roles, AddUserToRole, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=293) [of course you can see here there are other very similar methods to do things like add a user to multiple roles](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=304) [or add multiple users to a row or even multiple users to multiple rows, so there's lots of features built right](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=309) [into the framework. And of course then we could just simply say, we would like to add "johnsmith" and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=315) [we would like to add "johnsmith" to our Admin role. Now, of course, you wouldn't hard-code this; these would](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=321) [be values that would normally be provided by the user during the administration, but this gives you a good idea](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=326)[of how all of this is built into the framework and is really, really easy to access.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=332) [If you don't use these features, you do end up writing a lot of plumbing code yourself.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=337) [You'll end up writing your own data access and your own methods to insert users into roles and bind everything](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=342) [back up together and that's really not necessary when you're using the role provider.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=347) [The very last thing whilst we're looking at the facilities in the role provider is that that web application](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=353) [only showed us the Admin link when we're actually logged in and in the Admin role and that's very simple to do](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=358) [and it is implemented over in our Shared view. So, if we drill down into the layout, we will see that there's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=365) [a very, very simple condition which says if the user is in the role Admin, then display the hyperlink.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=374) [Now of course that's not an access control, but it is a usability control.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=380) [Having said that, there are many times where you may want to programmatically use a feature such as User.IsInRole](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=385) [and that was one of the points that was made in one of the earlier slides.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=392)[Sometimes you do need to get down to a very fine-grained programmatic approach and that's when you want to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=395) [pull out an option such as this and then build conditional logic around the roles that a user may be in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=401) [The very last thing to mention regarding the role provider is that really when we implement authorization systems,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=408) [we want to take a very role-centric approach and not a user-centric approach and the reason is simply for manageability.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=416) [We don't want to end up with a situation where we're coding in conditions, such as if the user equals troy hunt](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=424) [or if the user equals john smith, then allow them access to this or allow them access to that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=431) [It starts to infest the entire application with rules about individual people.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=436) [You're much better off being role-centric; if the user is an Admin, allow them to do this or allow them to do that,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=442) [and then manage that role membership somewhere else. It's a really fundamental access control principal and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=450) [it's very important that you do try and adhere to that, and avoid those individual username references.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=6&mode=live&start=457)

[Other access controls risk and misconceptions](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live)

[When developers build access controls into web applications, it's not particularly hard to identify which pages](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=0) [and which URLs need to be protected. They're very visible in the browser, it's very easy to test when a resource](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=8) [hasn't been properly protected. There are, however, increasingly a large number of resources that need to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=15) [protected that are harder to test and it's simply because they don't have the same sort of visibility.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=23) [So, for example, APIs, particularly APIs that are called either asynchronously via the web application or even](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=30) [via mobile devices, you never tend to see those APIs exposed in an address bar somewhere so the visibility of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=38) [them is actually quite low. Now just because it's not front and center in the developer's browser doesn't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=46) [mean that it's not possible to see which resources are being requested.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=53) [It's very easy to see, for example, when a button is clicked on a web page, there is an asynchronous request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=57) [to an API so it's very easy to see through Developer Tools in Internet Explorer or any of the other major](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=64)[browsers for that matter and it's also very easy to see through proxying tools, so something like Fiddler or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=70) [Charles, for example. It's very easy to monitor the requests, any HTTP requests, that the browser or the PC is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=75) [making and an attacker will look for those and then go and probe them.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=84) [So they'll attempt to do things like parameter manipulation to see if there's any insecure direct object references,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=88) [to see if there are resources being accessed where maybe there are not access controls in place and just by](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=95) [manipulating the structure of the request, they can cause the application to do things that it's not meant to do.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=102) [So it's very important that those APIs are properly protected and they're not neglected just simply because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=108) [they don't have the same visibility in the address bar as other web pages.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=114) [It's a similar story for other resources that are often not loaded in the browser, so for example, Word or PDF documents.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=121) [Often, a web application will need to generate these and in many cases, those documents can contain sensitive](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=129)[information, which is not meant for general public consumption; there needs to be authorization rules around](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=136) [those resources. It's a similar sort of story with things like data persistence in an XML file.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=142) [There are many cases where it's perfectly legitimate to store data in an XML file, but of course there's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=148)[nothing to stop an attacker making a request to that path and simply downloading the file directly off the server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=153) [Very often, these sort of resources simply sit on the file system, so they may not be part of say the response](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=160) [of a handler where you're actually requesting a dynamic page, which will apply authorization logic, generate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=167) [the resource, and then send it back in the response. So when those assets sit there on the file system,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=173) [they need to be able to benefit from the same sort of protection that the rest of the application has.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=179) [That's one of the great things that was introduced in IIS 7, the integrated pipeline.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=185) [So what it means is, is that the request for resources such as PDFs or Word documents, as well as things like](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=190) [JavaScript files and images and very common web page assets, all go through the same request pipeline and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=197) [the same authorization pipeline as a request to an ASP.NET web page or even an MVC controller.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=204) [What that means is, is that we can apply the same authorization logic, so using things like the location element](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=211) [in the Web.config to protect assets a very, very different nature.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=218) [So that's one option we have. The most important thing though is to recognize that there are file types that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=223) [are fundamentally different to web page assets that do still need to be protected and the right authorization](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=230) [rules must be in place, that's absolutely essential.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=236) [One of the things that does seem to cause some trouble is there are a number of misconceptions around what](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=244) [constitutes an access control. So, for example, sometimes you'll see developers creating paths that are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=250) [obfuscated so it'll be a URL with a very random ID. In many cases, it's not such a random ID.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=258) [In the Apple and AT&T case earlier on, it obviously wasn't a very random ID, but even when it is a very random ID,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=265) [you still need to supplement that path with a form of access control; there needs to be an authentication and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=272) [authorization process in place. Just believing that the resource won't be accessible because an attacker](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=279) [probably can't guess the URL is not sufficient and there are many cases of where URLs have been leaked through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=286) [all sorts of different ways and when there's a lack of access control, obviously there remains no protection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=293) [whatsoever for the resource. Another misconception I've heard a number of times is that if a website doesn't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=299) [have a domain, so it's only got an IP address, that that gives it some form of protection, that it will be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=305) [harder to find, that it won't be indexed. Of course, that's simply not the case.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=311) [Domains are great for usability so people can recognize the sites or they can easily type in the address,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=316) [but they don't imply any sort of lack of security, which you gain just by accessing a resource over an IP address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=322) [Certainly they can still be indexed by Google, they're still searchable, and of course IP addresses do adhere](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=329) [to a very discoverable pattern. And particularly with IPv4 addresses, there is a very limited range.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=335) [So standing up a resource on the end of an IP address is by no means an access control.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=342) [You also want to be very cautious about putting credentials in URLs.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=348) [I mentioned earlier on there are multiple points where URLs get leaked, so in other words they get disclosed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=353) [at certain points of the request response lifecycle. They also do things like get logged by proxies or gateways,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=358) [they get logged by web servers, and they sit in web server logs in plain text.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=365) [So you must expect that a URL is not a means of transporting sensitive data and, of course, credentials are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=369) [sensitive data. Now this also extends to HTTPS addresses as well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=376) [You can't assume that just because you're requesting a resource on the end of an HTTPS address that the URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=381) [won't end up cached at some point. So, again, I just mentioned web server logs; just because a request is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=389) [made of HTTPS doesn't change the fact that the web server by default will probably log that request somewhere](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=395) [and if that request contains sensitive data, you then end up with usernames and passwords and other sensitive data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=402) [sitting in web server logs. And there are other examples of where sensitive information in the URL can still be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=408) [captured, even with an HTTPS request. So whichever way you look at it, URLs are not the right place to put credentials](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=415) [and URLs alone, no matter how obfuscated or what data is passed through in them, are not a means of access control.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=423) [There needs to be an authentication and authorization process somewhere within that request pipeline.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=7&mode=live&start=431)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live)

[Let's take a look back over the module and probably the most important thing to recognize is that there are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=0) [multiple ways that we can implement authorization within ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=5) [The first thing we looked at was Web.config locations and whilst that made things very manageable in terms of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=11) [having all your access control roles in one place, clearly that was problematic once we got into that ASP.NET MVC](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=17) [application and we found that we actually had multiple routes through to the same resource, and without multiple](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=24) [location elements and the appropriate authorization within in them, those additional routes actually circumvented](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=29) [the access control. We moved onto looking at the authorize attribute on the controller or on controller action](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=36)[and as we saw with the ASP.NET MVC application, putting it on that entire Admin controller gave us protection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=43) [for the resource regardless of what routes were actually mapped through to it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=50) [So the authorize annotation is an extremely valuable way of protecting the MVC web app.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=55) [And then, of course, we also looked at some of those features within the role provider, such as User.IsInRole.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=61) [So we can always implement that as an authorization control around a particular piece of logic, such as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=67) [whether or not we display a link to the Admin page. What this really boiled down to was whether we were securing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=74) [a path or securing a resource. Securing a path works great when that path always maps through to the one](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=80) [physical resource. So, for example, in an ASP.NET web forms application, if you secure the Admin path, you're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=88) [going to have all of your .aspx files sitting in the Admin path, they'll all be protected.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=95) [In the MVC application, really the controller is the resource that we want to protect and just protecting the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=101) [path wasn't sufficient. We looked at role-based authorization with the role provider and how easy it is to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=107) [build into an ASP.NET application. Now, of course, once that's built in, you can either jump into the database](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=115) [and map users to roles or you can use those features, such as Roles.AddUser and work with it programmatically](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=122)[without actually having to write up all the data access plumbing and there are a whole bunch of different](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=129) [methods within the role provider that make working with users and working with roles extremely easy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=134) [And, again, one of the things I mentioned at the end of that part of the course is be very, very conscious about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=140) [using roles and applying permissions on a role basis and not doing it at an individual user basis.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=145) [The manageability of it just gets extremely difficult when you do that, so be role-centric.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=152) [And finally, there are some of those little "gotchas" with access control, so particularly remember things like APIs.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=159) [We're increasingly moving into an asynchronous web application world.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=166) [Things like single page applications are fantastic, but don't forget to protect the resources behind those](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=170) [asynchronous requests. And, of course, it's the same story if you're building APIs for mobile applications --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=177) [make sure appropriate authorization is happening within the API itself.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=183) [Lastly, we looked at trying to remember that really any publicly facing resource should be considered publicly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=189) [accessible - so things like obfuscation; security through obscurity and not access controls, simply standing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=196) [up a resource behind an IP address is not an access control; putting credentials in a URL is risky, even over HTTPS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=204) [All of this comes back to the fact that if you're making a simple HTTP request for a resource without having](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=214) [authenticated and that resource is returned without an authorization process, it is very, very likely that you](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=221) [have failed to restrict URLs.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m8-url-access&clip=8&mode=live&start=228)

[Insufficient Transport Layer Protection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to Part 9 of the OWASP Top 10 on Insufficient Transport Layer Protection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=0) [Throughout this module, you might also see it referred to as TLS, SSL or HTTPS, and whilst there are subtle](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=7) [differences between all of those, the terms do tend to get used a little bit interchangeably.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=16) [In this module, as with all modules, we'll start out by taking a look at how OWASP views the risk and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=23) [then we will move on to performing an attack against yet another vulnerable application and this time,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=30) [we're actually going to be sniffing insecure traffic sent across a hijacked wireless connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=36) [We'll then move on to taking a look at the concept of secure cookies, what the risk is when they're not secure,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=43) [and how we can properly secure them within ASP.NET. We'll then take a look at how we can force both web forms](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=50) [and ASP.NET MVC to always use a secure connection for certain resources, simply because there are some resources](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=57) [which we never want to load insecurely. We'll take a look at the risk of mixed mode content when nonsecure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=66) [resources are loaded into a secure page and then we will move on to looking at HTTP-strict transport security](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=73) [or HSTS so that we can force requests to be secure for a particular site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=81) [And finally, we will take a look at some HTTPS anti-patterns and have a look at some of the other considerations](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=88) [you need to keep in mind when planning HTTPS for your web application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=0&mode=live&start=95)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live)

[Let's take a look at the OWASP overview and risk rating and the first thing to consider is that when we look](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=0) [at Threat Agents, that could be anyone who is able to monitor the network traffic of your users.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=7) [Particularly when we're talking about a website, there are many different ways that your users may connect](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=14) [to that website and we need to consider how that traffic is protected when you consider that the connections](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=20) [could range from a wide desktop connection in a secure corporate environment all the way through to an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=26) [internet café with an open wireless connection. So there are very, very different risk profiles.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=31) [The Exploitability is classified as difficult simply because it can be difficult to monitor network traffic,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=38) [but as we'll see in a moment, it can also be exceptionally easy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=45) [The primary thing when we talk about insufficient transport layer protection of websites is that that monitoring](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=49) [most frequently happens somewhere close to the user, so for example, on an insecure wireless connection,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=56) [but of course it could happen anywhere between the user's browser and the server running the website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=62) [So there really are a lot of different options and we will look at some of the places where network traffic](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=67) [is at risk throughout this module. The Prevalence is classified as common simply because there are so many](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=72) [different ways that SSL or TLS can be implemented insecurely and indeed this risk itself is classified as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=80) [insufficient transport layer protection, which indicates that in many cases there may be some degree of HTTPS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=89) [used for a website, but very frequently, aspects of the implementation are vulnerable.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=97) [However, it is very easy to detect and we'll look at some of the ways we can identify that throughout the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=103) [rest of this module. The Technical Impact is really going to be very dependent on the nature of the data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=109) [that hasn't been properly protected. We're going to look at an example of session hijacking where the risk](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=117)[will then be an attacker impersonating the victim, but of course, insufficient transport layer protection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=122) [could also mean not properly protecting things like credentials or even credit cards or other financial data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=129) [So the impact can be quite significant, but it's very dependent on the nature of the data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=135)[Naturally the Business Impact will be dependent on the information which an attacker may gain access to.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=141) [An important point that's made here though is that when we talk about the risk, we're talking about both](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=148) [confidentiality, so has an attacker been able to gain access to the data, and integrity, so has an attacker](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=154) [been able to change the data, and that's extremely important because many times people think of SSL as only](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=160) [being about encryption in order to protect the data from prying eyes, but it also offers an extremely valuable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=167) [proposition, which is integrity, so ensuring that an attacker cannot change the data and we'll look at some](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=175) [ways that they might want to change it in just a moment. Let's move on to actually exploiting a vulnerable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=182)[application because that's the really interesting side of insufficient transport layer protection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=1&mode=live&start=187)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live)

[In order to demonstrate the risk of insufficient transport layer security, we need to first understand what](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=0) [a ‘man in the middle' attack is, simply because what I'm going to do in this demonstration is show what happens](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=8) [when someone else is able to intercept traffic that is not encrypted, that does not have sufficient transport](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=14) [layer protection. When we talk about a man in the middle attack, we often see it abbreviated as an MiTM attack,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=23) [and really what this means is we have someone who makes a request for a resource, so often that will be a request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=31) [for a web page via their browser. Under normal circumstances, the server receives the request just as the user](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=39) [issued it and it then returns a response and, again, under normal circumstances, the user will receive the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=46) [response just as the server returned it. When everything goes to plan, nothing is manipulated, nothing is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=53) [observed, it's simply packets flying backwards and forwards between user and server in a private manner.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=60) [When we bring a man in the middle into the picture, we now have a third party who sits between the user,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=67) [who will now start calling the victim, and the server. Now the man in the middle can either observe or manipulate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=74) [any traffic that is sent insecurely across that connection. So, for example, they can look at the data in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=82) [request, so things such as request headers or any form data that might be posted.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=90) [They could then look at anything in the response returned from the server, so such as the headers and such as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=95) [the body, which might be in the HTML or other content that appears on the web page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=101) [It's also important to remember that the man in the middle can manipulate the traffic, so they could change](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=106) [the nature of the request or they could even change what is in the response.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=111) [So, for example, they could inject JavaScript into the response body and we'll have a look at a precedent of that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=116) [a little bit later on in the module.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=122) [There are many different ways of getting a man in the middle between the victim and the server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=126) [So, for example, a very rudimentary way would be to use a physical wiretap, which is nothing more than a device](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=132) [that sits somewhere on an Ethernet cable between the victim and the server.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=139) [That is one way of getting a man in the middle and monitoring or manipulating the traffic.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=144) [A man in the middle could also be at the ISP level. So obviously when a user connects to the internet,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=148) [they usually do so through an internet service provider and obviously there is traffic that goes backwards and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=154) [forwards through that provider. When there is insufficient transport layer security, that provider can, again,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=159) [observe or manipulate the content. It's also possible to monitor unprotected traffic at a Wi-Fi hotspot.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=166) [So this is why free connection, such as you often find at say a coffee shop or an airport, are so risky --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=173) [because it's usually possible to see what data is being sent over that connection from other users when it is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=179) [not properly protected by the website they're using. Another approach is to create a rogue wireless access point](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=186) [and then have victims connect to that access point and monitor their traffic and that's what I'd like to do now](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=195) [to demonstrate the risk of insufficient transport layer security.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=200) [Now the way we're going to do this is we're going to use what's called a Wi-Fi Pineapple.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=204) [This is just a little device around about the size of a cigarette packet with an antenna.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=210) [Let's take a look at how we're going to use this to exploit a website with insufficient transport layer security.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=216) [So first of all, we are going to have a victim and in this case, the victim is going to be my Surface Pro.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=225) [The victim is simply going to browse a website that is not properly protected and then we'll see what the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=232) [attacker can do with that. So let's bring the attacker into the picture and the attacker is the machine](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=237) [that is running a Pineapple. What's going to happen is the victim is going to connect to the Pineapple,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=244) [which again is running a rogue wireless access point so the victim thinks it is just connecting to a normal](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=249) [WiFi network, whereas in fact, the Pineapple is then going to route that traffic through an attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=255) [The attacker is going to monitor that traffic and then send it out via its own WiFi signal out to a router](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=262) [and across to the internet. What it means is that the victim just has a perfectly normal browsing experience.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=269) [They are absolutely none the wiser that we have put a man in the middle between their WiFi connection and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=276) [the internet itself and this is really, really important to see because it helps you understand that to the victim,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=282) [it's business as usual; they have no idea that there is someone in the middle monitoring their traffic and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=289) [that's why attacks like this can be so effective and why transport layer security is so important.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=294) [Let's now move on and execute this attack.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=301) [In this demo, we're actually going to see two different machines.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=306) [The first machine we're going to look at is the attacker and then as we start the attack, we'll also see the victim,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=311) [but what you see in front of you now is the attacker running a piece of software called Wireshark.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=317) [Now what Wireshark does is it analyzes network traffic and we're going to use it to monitor and then inspect](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=323) [the network traffic that runs through the attacker's machine. So if you recall, the attacker is connected to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=330) [the WiFi Pineapple, which is a rogue access point, which the victim will then send its traffic through.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=336) [That WiFi Pineapple is connected to the Intel Gigabit Network Connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=341) [So, what we're going to do is just select that and then we will go down to the Capture Options, leave all the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=346) [defaults, and Start the capture. Right, so what we see now is Wireshark is starting to capture all of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=353) [packets that are running through that network adapter, so running from the WiFi Pineapple and then through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=362) [the attacker's machine. There's actually a lot of traffic that goes through a network interface that we're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=367) [not interested in. So what I'm going to do here is just add a filter so that we only monitor HTTP traffic,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=373) [because obviously HTTP is the language browsers are speaking when they're loading websites.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=380) [So let's just add that filter and now we see all of the other traffic disappear. So that's great.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=385) [We're only going to see data appear now when it is an HTTP request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=391) [So this is the attacker set up and ready to go. What I'd like to do now is introduce the victim and I've](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=396) [recorded this in parallel on my Surface Pro. Now the victim is already connected to the rogue access point](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=403) [that is the Pineapple. The victim also has our target website open, so this website is running on a server,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=411)[it's a totally different machine to either the attacker or the victim, and what we can see is, again, a very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=418) [standard ASP.NET template and we can see that the victim is already logged on - we can see ‘Hello troyhunt'](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=425) [there towards the top of the page. Now what I'd like to do in this demo is illustrate the risk of unprotected](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=431) [auth cookies and at multiple times throughout this course, we have looked at what auth cookies are.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=439) [Now we're going to see what happens when they're transmitted insecurely.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=446) [Now to do this, I'm going to make it extremely simple and all I'm going to do on the victim is just follow one link.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=450) [So I'm going to click on the About link and as I click on that, you can see to the left of the screen that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=458) [the attacker is now watching those packets. So we can see a couple of rows appear there, which are filtered](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=464) [HTTP rows. They're only the HTTP requests - that is the request that the victim has just issued, and we can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=470) [also see the response that was returned from the server. Now that is actually all we need from the victim,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=477) [just that single request. So we can now move that off the screen and let's start focusing on the data that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=483) [the attacker has collected. So we can see here in this first record, it was a GET request to /Home/About,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=490) [which is correct, and if we right-click on that record, we are able to follow the TCP stream and what that's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=497) [going to give us is the actual bytes in the request and response which we just witnessed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=504) [The really important piece that I'd like to look at though is the request headers and we can see here in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=510) [the request header there is a cookie called .ASPXAUTH and, again, we've looked at this multiple times.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=516)[This is the way that authenticated session is persisted across individual requests.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=523) [What I want to do is hijack that session and the way that I'm going to do this is to first of all simply](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=530) [copy the cookie value. So we'll select all that, Copy, and now we have it in our Clipboard.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=536) [To hijack the session, I now need to go and recreate that cookie in my browser and to do this I'm going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=543) [jump over into the Google Chrome browser because there's an extension available that makes this very easy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=550) [The first thing I'd like to point out here is that we have the target site loaded, but clearly I'm not logged on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=557) [and just to make sure, I'm just going to give it a Refresh and we can see that I still have a Login link on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=564) [the top right-hand corner of the screen. It's not giving me a Welcome troyhunt so clearly I am not logged in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=569) [to this website and if we follow a couple of links, again, nothing changes -- we still have the Login link.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=576) [Now what I'm going to do is use a little extension called ‘Edit This Cookie' and the reason I'm going to use](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=583) [this extension is because it makes it very easy to recreate cookies.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=589) [There are other extensions and other means of doing the same thing in different browsers; I just find this one](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=593) [quite easy. So, what I'm going to do is add a new cookie and I'm going to call it .ASPXAUTH and now I'm going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=598) [to paste in the cookie that we captured via Wireshark. So, again, this is the cookie that the victim was](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=606) [sending to the web server. I'm going to mark that cookie as HostOnly and HttpOnly so it's consistent with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=613) [the way the web server would normally set it and now I'm going to submit cookie changes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=620) [So, that cookie is now set in my browser. Let's reload the page and here we go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=626) [I am now logged on as troyhunt, which means that we have successfully hijacked this session.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=633) [Now note that this doesn't mean we have the credentials of the victim, but it does mean that we have now taken](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=639) [their identity, insofar as we can do whatever the victim could normally do while they are logged on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=646) [This is exactly the equivalent of the victim logging on to a website, walking away from their PC, and then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=653) [an attacker sitting down and using that website on their PC. We've just done it by sniffing packets and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=660) [recreating the cookie and that is exactly what the risk of insufficient transport layer security is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=667) [When sensitive data is sent unprotected and an auth cookie is clearly sensitive data because it allows us to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=675)[hijack the session, this is what the risk is and this is why transport layer security is so important.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=2&mode=live&start=681)

[Risk in practice: Tunisian ISPs](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live)

[There have been many, many precedents of where insufficient transport layer protection has led to the disclosure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=0) [or the manipulation of data transiting a network, but a particularly poignant example is the Tunisian government](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=7) [harvesting usernames and passwords back in 2011. Now during this time, there were a lot of political uprisings](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=15) [in Tunisia and around this time, it was discovered that there was injected JavaScript that captures usernames](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=23) [and passwords on popular sites, such as Gmail, Yahoo, and Facebook.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=30) [So what was actually happening here is that web applications that did not have sufficient transport layer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=37) [protection were having script injected into the Login pages. So this wasn't so much an issue of reading](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=42) [credentials or in other words the disclosure aspect of this risk, this was more about what we mentioned in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=49) [the overview and risk rating, where it's an issue of integrity - the content of the page was actually changed.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=56) [As we scroll this article, we get a little bit more of an idea about the execution of this particular attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=64) [and, in fact, what we can see here is that the script that was injected into these pages would pull the username](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=70) [and password when the user logged on, encoded it, and those encoded credentials were then sent with a GET request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=76) [to a URL. The fact that it was noted as being a non-working URL doesn't really matter because if it's an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=85) [HTTP request and an attacker owns the network that the request is being sent over, then they can interpret it anyway.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=90)[The other thing that's really significant with this particular attack is the fact that it does appear to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=98) [the Tunisian government, which was mounting it. Because the government owns the ISPs and because they can then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=103) [monitor anything that transits through those ISPs, they had the power to not just spy on their citizens,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=110) [but actually change the content of the pages that local Tunisians were downloading.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=118) [Of course, what's really significant about all this is that in cases where HTTPS was used, there was not a risk](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=124) [and, in fact, the article makes quite a point of saying that in each test case we were able to confirm that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=132) [Gmail and Yahoo were only compromised when HTTP was used, because of course HTTP offers no transport layer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=136) [protection by default. It's only when we have an HTTPS connection that we have that confidence of both](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=146) [confidentiality and integrity. The other thing that's really worth well noting here is that these services](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=152) [were actually posting credentials over a secure connection. So even though the Login page was loaded over HTTP,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=160) [when the user clicked the submit button, their credentials were sent over a secure connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=167) [The problem, of course, is that because the Login form itself was loaded insecurely and there wasn't the assurance](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=173) [of integrity, that Login page could be manipulated and the injected JavaScript effectively just became a key logger,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=180) [which could capture the information that users entered and then send it off in parallel to then legitimately](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=187)[submitting the form and submitting their credentials over a secure connection to the actual site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=193) [So this is a very good example of that insufficient component of this risk --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=199) [yes there was HTTPS, no it was not enough to protect their users.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=3&mode=live&start=204)

[Demo: Understanding secure cookies and forms authentication](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live)

[Let's move on now and take a look at an app which is vulnerable to insufficient transport layer protection and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=0) [what I'd like to look at first is the risk of unprotected cookies.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=7) [As we saw in the attack we mounted earlier on with the Pineapple, cookies are actually extremely valuable](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=11) [little pieces of data. Many people think it's only the credentials that need to be protected, but as we saw,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=17) [when cookies aren't protected and particularly when an auth cookie is not protected, there is a significant](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=24) [risk posed because an attacker may be able to hijack a session.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=30) [Fortunately, cookies have a built-in native feature regardless of the server-side platform, which will prevent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=35) [them from being sent over an insecure connection. So let's take a look at what that looks like in ASP.NET.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=41) [So the way I'd like to demonstrate this risk of insecure cookies is to start by using our sample insecure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=49) [application and as usual, it's just a very standard ASP.NET MVC 4 web application; it's just got a couple of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=55) [little customizations. So let's start by hitting F12 and turning on the Internet Explorer Development Tools](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=63) [and then I'm going to jump over onto the Login page and before I log in, I would like to start capturing the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=70)[request because I want to show you what is going to be in one of these response headers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=76) [And I will log on with a sample account and with a sample password and here we go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=81) [So, what we can see here is we can see a request to Account.Login and it is a POST request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=89) [Let's double-click on that and take a little bit of a look at the cookies that were set here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=95) [Now we've seen this .ASPXAUTH cookie many times now, that was the one we hijacked earlier on with the Pineapple.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=101) [What I'd like to point out here is that this response showed that the browser has received that .ASPXAUTH cookie](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=108) [and one of the things that we'll note here is that Secure is set to No.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=116) [Now this is quite important because what it means is, when the browser makes subsequent requests to this website,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=121) [the cookie will be sent whether the connection is secure or not. Now, of course, if this cookie gets sent via](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=127) [a request to a nonsecure resource, it is at risk of interception and, again, that's what we saw a little bit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=134) [earlier on. If the cookie was flagged as secure, it would only be sent with the request to a secure connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=141) [Now there is a very, very easy way of changing this in ASP.NET so let's jump over to Visual Studio and see how that's done.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=148) [Okay, so over in Visual Studio and the first thing we want to do in order to make sure that our cookies are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=158) [flagged secure when we log on is go down to our forms node. Now, of course, we're using the membership provider](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=164) [in this ASP.NET MVC application so that's just the default implementation and that has a dependency on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=171) [forms authentication. Now what we can do with forms authentication is we can set a flag to require SSL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=178) [Now by default, that is false, which means that forms authentication will work over an insecure connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=186) [We are going to make that true because we want to make sure that you cannot send the authentication cookie](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=194) [in an insecure fashion where we could have the problem we saw before when it got hijacked.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=200) [So, let's just save that and we'll jump back over to Internet Explorer and see how this changes the experience.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=205) [Okay, so back in IE, we can just remove those Dev Tools for the moment, log ourselves off, and then we will go](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=213) [and log ourselves back on and see what happens. So, same credentials as before and the password,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=219) [and now we get an error and the error is really pretty self-explanatory -- the application is configured to issue](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=229) [secure cookies, that's what they require SSL flag did. These cookies require the browser to issue the request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=235) [over SSL using the HTTPS protocol. Now, of course, we have actually loaded this page over HTTP and that is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=242) [why we are seeing this error. So what I'm going to do is just change the scheme so that we are using HTTPS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=251) [Let's go back to the Homepage and we will try and log on again.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=258) [One of the first things you'll notice about this page is that the address bar has now gone red.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=262) [Not only is it red, but we can see a great big notification here, which says there is a ‘Certificate error'.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=268)[Now if we drill into this and we have a look at the Certificate error, we can see that the security certificate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=274) [presented by the website was issued for a different website's address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=279) [Now if we take a look at the certificate, we can see that it was issued to TroyPC by TroyPC.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=283) [This is a self-signed certificate, so it's a certificate that's good enough to use here locally while I'm doing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=290) [some development, but of course it's not suitable for launching a website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=296)[You need to get a certificate issued by a certificate authority or CA in order to actually stand a live website](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=300) [up behind it. Now that is an absolutely fundamental component of secure certificates and of HTTPS SSL TLS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=308) [The concept of the browser validating that the certificate has been issued by a trusted certificate authority](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=318) [is an absolute cornerstone to HTTPS. We're not going to go into the details about the implementation of that,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=326) [but it's worthwhile simply knowing that certificates are issued by certificate authorities and certainly if](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=334) [ever you see warnings like the one here about a certificate error, do not trust the site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=340) [Okay, so let's move on. Now interestingly what you'll see now that we're back on the Homepage is that I am](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=346)[actually logged on. So even though we got that exception about not using HTTPS, we did still manage to log on,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=352) [but of course that's not a very good experience. So let's just log off and what I want to do is open the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=359) [Developer Tools again because I'd like to show you what happens when we log in when SSL is required.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=365) [We'll go to the Login page, start capturing, and let's log in again with the usual username and the usual password](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=371) [and have a look at what happens. So, of course, we get a normal request, we get a request to the Account/Login path,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=381) [and I'm going to open that up, and what we will see now is that there was a cookie received, so that was sent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=388) [to us via the server, and the Secure flag is now set to Yes. Now this is very, very important because it means](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=395) [that this cookie will not travel with a request to an insecure resource.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=403) [So let's have a look at what that looks like and what I'm going to do is just clear the existing request from](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=408) [the Developer Tools and if we go up and have a look at the About link, down in the Status bar at the bottom left](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=414) [of the screen, you can see that About is actually a link to an HTTP address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=420) [Now just before I click that, we can see in the top right-hand corner that I am definitely logged on, it says](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=426) [‘Hello troyhunt', let's now follow that About link and suddenly you can see that that top right-hand corner](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=431)[is now asking me to log in. So we're now on an insecure page and it doesn't know who I am.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=438) [Now we can see why that is if we go down and have a look at the request in the Developer Tools.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=445) [So double-click to open and what we'll see is that nowhere here was that .ASPXAUTH cookie sent.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=451) [There were two other cookies, but not the ASPXAUTH cookie and it wasn't sent simply because it requires](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=458) [a secure connection to send. If I go back and hit this via an HTTPS address, the cookie is still there,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=465) [so if I clear the traffic and I go and request this resource via HTTPS, we can see that I'm now logged on again,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=471) [‘Hello troyhunt' is back there; if we go and have a look at the request, we'll also see the cookie was sent,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=480) [simply because it is a secure cookie. You don't see it flagged as secure when you send the cookie with the request,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=486) [you only see it when it is sent back to the browser from the server, but certainly it is secure and we've just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=492) [demonstrated that. The reason that secure cookie is so important is not just so that if you never get back](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=498) [to an HTTP address, it won't be sent, but it's to make sure that even a very inadvertent request to an HTTPS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=505) [resource, so for example if a develop embeds an image on the page and the image source is the same domain](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=513) [on an HTTP scheme, the cookie won't be sent. So make sure that there can be absolutely no accidental risk](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=521) [of that cookie being sent insecurely and that's a very, very powerful position to be in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=4&mode=live&start=529)

[Demo: Securing other cookies in ASP.NET](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live)

[Now auth cookies are one thing, but of course there are many other use cases for setting cookies in the browser](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=0) [and very often we want those to be secure as well and there's another example I'd like to show you that we can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=6.5) [easily see by turning the Developer Tools back on, starting our capture, and just going back to the Homepage](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=12.5) [over the HTTP scheme. So we'll take out the path, go back to there, have a look at the request that's issued,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=19.5) [and here we can see a request verification, cookie was sent. Now if you recall, back in Part 5 of the series](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=27.5) [when we looked at Cross Site Request Forgery, the request verification token was sent as part of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=35.5) [synchronizer pattern and then validated by the MVC controller to make sure that the request wasn't sent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=41.5) [maliciously by an attacker, so the attacker wasn't tricking the browser into making a request on its behalf,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=48.5) [but clearly in this case, that cookie has been sent across an insecure connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=55.5) [So, of course, it does pose a risk to a man in the middle attack. So what I'd like to do now is make sure that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=60.5)[this cookie is set securely so let's jump back into Visual Studio and make that happen.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=66.5) [Here's where we left off before - where we set forms authentication to require SSL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=72.5) [Now, of course, this is only going to apply to the auth cookie. What I'd like to do is make sure that all cookies](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=78.5) [set by this website are secure and the way we're going to do that is to simply drop down in system.web and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=85.5) [add an HTTP cookies element and within HTTP cookies, we can access the same property as we did in forms auth,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=92.5) [which is simply require SSL and we set it to True. That's all it takes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=100.5) [Let's save that and then we'll jump back into Internet Explorer and see how things change a little bit.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=105.5) [Back in Internet Explorer and what I'd like to do, before I demonstrate this, is just clear all the cookies](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=112.5) [that we already have in the browser. So I'm going to go to Cache in the Developer Tools, I'm going to clear](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=117.5)[all the session cookies, and just to be sure, we will also clear all the cookies for the domain.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=124.5) [So, let's get rid of all those and we can get the Developer Tools out of the way for the moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=130.5) [Now what I'd like to do is try and log in. Now when I log in, we're going to see something a little bit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=135.5) [different happen. So, let's try and log in with my User name and the same old Password and log in and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=140.5) [here we have a very different error. Now this time we're missing the required anti-forgery cookie.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=148.5) [We saw a very similar error back in Part 5 when we talked about CSRF and, again, if you recall, the anti-forgery](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=155.5) [protection works by submitting both the hidden field with a token and a cookie with a token.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=163.5) [Now in this case, what's happened is that the cookie wasn't present and, of course, the reason this is,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=168.5) [is because all of our cookies now require SSL. This is not an SSL connection, therefore the cookie was not sent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=174.5) [with the request and consequently our anti-request forgery protection has thrown the error you see in front of you.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=182.5) [So, in fact, what we've just done is ensured that every time we decorate a controller action with a validate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=189.5) [anti-forgery token annotation, that controller action must be requested securely.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=195.5) [Now there is still a little bit of flexibility in the way we flag cookies as being secure or not, even when we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=202.5) [set require SSL to be true on the http cookies element of the Web.config.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=209.5) [There are valid use cases where you may have a site with many secure pages, but you still want to have a cookie](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=215.5) [that is not flagged as secure. So, for example, you may want to persist something very, very simple and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=221.5) [non-sensitive just such as a name and you might want to persist that across both a secure connection and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=228.5) [an insecure connection. So perhaps you could be browsing the insecure part of the site, loading it up without](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=235.5) [needing HTTPS, but still saying Hello troy, welcome to our site, and then when you needed to do anything of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=240.5) [a secure nature such as access an auth token or maybe provide some financial data, you would then access](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=247.5) [an HTTPS address and that's the only way that you'd get any cookies of a secure nature.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=253.5) [So there can be this dichotomy where you want to have a cookie that persists across both secure and insecure schemes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=259.5) [Now if we jump back to the Homepage and we open up the IE Dev Tools again and we start capturing and we reload](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=266.5) [this page and have a look at the first response, we will see that there is a cookie that has been received and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=276.5) [this cookie is called MyCookie with a Value of My cookie value and it is not secure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=284.5) [Now this is even though we have set RequireSSL onto HTTP cookies.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=290.5) [Let's jump over to Visual Studio and see how this was done.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=296.5) [Back in Visual Studio, we can see that the Index controller action, which is sitting on our HomeController,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=301.5) [so this would be the root of the site, is setting a cookie into the Response and, of course, this is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=306.5) [cookie that we were just looking at; it's named MyCookie, it has a value of My cookie value, but what's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=312.5) [important here is that it has the Secure attribute of the cookie set to False.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=318.5) [Now the reason why this is important is this context is that this allows us to override the RequireSSL attribute](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=324.5) [on HTTP cookies. So really, what that attribute in the Web.config is doing is it's changing the default.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=332.5)[When we don't set RequireSSL, it will default to False, which means that any cookies will default to not being](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=339.5) [secure; when we do RequireSSL, it changes the default to True, which means that every time we set a cookie and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=347.5) [we don't declare a Secure attribute, it becomes Secure. This is simply an override of that default attribute.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=354.5) [So really, what we've done with this application is we've moved it into the more secure by default position.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=362.5) [We've set it so that our cookies are secure unless we're very explicit when they're created and we tell them not to be.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=368.5) [So particularly, if you're building a web application where you are predominantly going to work over secure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=376.5) [connections, this is a good state to put it in because it means that things are going to be secure by default,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=382.5) [which is always a better position to be in and then if you don't like it for one particular cookie,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=388.5) [jump in and explicitly set it to not be secure.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=5&mode=live&start=393.5)

[Demo: Forcing web forms to use HTTPS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live)

[There are many instances when you're building an application and there is a requirement for SSL where there](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=0) [are certain pages, which you never want to allow to be loaded over anything other than a secure connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=7) [So, for example, the Registration page, which normally requires a password as part of the signup process,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=14) [must always be secure. The Login page must always be secure because that's when you're providing that username](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=21) [and password. So it's really important to make sure that these resources are only ever available over a secure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=27) [connection and you can't load them over HTTP, even if someone explicitly never gets that address and circumvents](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=34) [any links that you might have to the HTTPS scheme. Now fortunately we can do that very easily in ASP.NET and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=41) [the way we do it in web forms is a little bit different to MVC so let's start with web forms and we'll start](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=49) [by taking a look at a demo web forms page, which I've got within this ASP.NET MVC application.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=54) [So now we'll just follow the link over to the web forms test and, of course, the first thing you'll notice is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=61) [that we don't have any branding on this site, it obviously isn't using the shared view, which is part of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=66) [MVC site, but the important thing is, is that we can see here -- is this a secure request, False.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=71) [Now alternatively, if we change the scheme to HTTPS, we should see True.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=78) [Now this is going to give us an opportunity to start looking at how we can detect whether the request is for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=85) [a secure resource or not and then handle it accordingly. We can detect whether that connection is secure or not.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=91) [Let's jump over to Visual Studio and see how that's done.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=98) [Over in Visual Studio, we can see that this is just a very, very simple page and the detection of whether that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=102) [connection was secure or not is simply done by accessing the IsSecureConnection attribute of the Request object](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=109) [and that's all it takes; it's extremely simple. So we can now use that to detect when the connection is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=117) [insecure and change that so that we can redirect to a secure connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=124) [So let's jump into the code behind of this page and let's just start out by detecting if the request is secure or not.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=128) [So, again, we'll just access that IsSecureConnection property and, of course, what we want to do is change things](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=137) [a little bit if the request is not secure and what I'd like to do is simply define a secure URL and the way](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=145) [we're going to access this secure URL is we're going to take the request and we're going to grab the URL property](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=152) [of that request, that is a type of URI so we're going to turn that into a string, and then we're simply going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=160) [to replace one scheme with the other, so we're going to take the insecure scheme, which of course begins with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=167)[http://, and we're going to change that to the secure scheme, which will begin with https://.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=173) [So that is our secure URL. Now in this example, we never want this page to be loaded over an insecure connection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=182) [so what we're going to do is we're going to do a Response.RedirectPermanent, which is just a little bit different](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=189) [from a Response.Redirect, which is what most people are probably familiar with.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=197) [A Response.Redirect will issue an HTTP 302, which means the resource is moved temporarily.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=201) [A RedirectPermanent will issue an HTTP 301, which is important because it actually means that the response has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=209) [moved permanently -- that old path no longer exists, this should not be an HTTP scheme to this URL, and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=216) [all we need to do now is just pass that secure URL variable.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=225) [So let's give that a rebuild and we'll have a look at how this looks in the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=229) [Back in the browser, let's start by going back to the Homepage of our test website, so we'll just drop off that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=235) [path and we will also drop off the HTTPS scheme, so we'll go back to HTTP.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=240) [Now what I'd like to do is fire up the Developer Tools again so we can see what happens, we'll start capturing,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=247) [and now let's go to our web forms test link, and here we go. So, what we can see first of all is obviously](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=253) [we've ended up on the HTTPS scheme again, we can see that in the address bar, we can see that the connection is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=261) [now secure, and down in the network traffic we can see that there was a request issued to TestSSL.aspx,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=266) [it responded with a 301, and again that 301 is a permanent redirect.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=274) [If we drill into this and we have a look at the response headers, what we're going to see is that there's also](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=279) [a location returned. So when we get that 301 moved permanently, it comes with a location as well, which is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=285) [where the browser should then make a subsequent request to. So what we've just done here is that we've forced](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=292) [this page to only load oversecure connection and do so by telling the browser that the resource has moved](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=298) [permanently to that secure location. So this is a very powerful feature and we've implemented it in a very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=306) [basic fashion. In many cases, you may actually want to create an HTTP module, which may even do this across](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=312) [the entire site. You might also want to move it up the page lifecycle a little bit, so doing it on page load](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=319) [is maybe a little bit too late in some cases so you can look at moving this up a little bit earlier.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=325) [The only other thing worth noting whilst we're talking about accessing that IsSecureConnection attribute is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=331) [that you do need to test this carefully in your target environment because there are some places where that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=337) [attribute may actually be misleading. So, for example, I've seen Cloud environments where the SSL certificate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=343) [sits on a load balancer in front of the web application. By the time the request actually reaches the web server,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=350) [it's no longer secure so there's this little gap internally within the host between that front-end and the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=357) [web server where the connection is insecure. What it means is, is that even though the request may have been](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=364) [issued securely and been sent across the internet in an encrypted fashion, the web server doesn't actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=371) [recognize that and in cases like that, you'll normally find that the device in front of the web server will then](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=378) [pass on a header, so something like an HTTP X-Forwarded-Proto header, and you then have to implement your](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=384) [application logic around the specific request header as opposed to the native IsSecureConnection attribute.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=391) [So just be cautious of that -- you do need to test that in your target environment. So that's web forms.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=398) [Let's go and take a look at MVC because it's a little bit neater when we apply this at a controllerAction level.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=6&mode=live&start=404)

[Demo: Requiring HTTPS on MVC controllers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live)

[Just as we saw in the web forms example, there are times when it's very important to make sure that a resource](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=0) [is only ever loaded over a secure connection. Now there's a really good example in this website and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=7) [we can see that by going to the Register link.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=13) [Now clearly the Register page is going to handle sensitive data; it is going to handle a password.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=16) [This page should never be loaded over HTTP. Even if it does post to HTTPS, which by default it won't,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=23) [but even if it did, this is still insecure and, in fact, this is the example that we looked at earlier on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=32) [with the Tunisian government. Once this page loads over HTTP, we cannot verify the integrity of it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=39) [For all we know, a man in the middle has already injected some JavaScript in this page to harvest the credentials](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=47) [as they are entered into the form. We can't trust this page. Let's make sure that it can only ever be accessed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=55) [over a secure scheme and to do that, we're going to jump back into Visual Studio.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=62) [Okay, back in Visual Studio and we're looking at the AccountController and, of course, the AccountController is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=68) [one of the default controllers that is in a new ASP.NET MVC application and it's the AccountController which](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=74) [loads the registration form. So, when we scroll down a little bit, we will see that there is a controller action](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=81) [called Register. Now that's the page that we were just looking at.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=87)[This is the action that's fired when we request /Account/Register.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=92) [We must make sure that this controller action can only ever be requested over a secure connection and with MVC,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=98) [it's really, really easy. All we need to do is drop down and add a RequireHttps attribute, that is it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=106) [So let's rebuild the application and jump back into Internet Explorer and see what happens now.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=114) [Alright, so we've now set our Register controller action to RequireSSL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=122) [So let's open the Developer Tools, start capturing the network traffic, and we'll just hit that Register link](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=127) [again, which of course is to an HTTP address -- and here's what happened.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=134) [So, we can see that we are now in the HTTPS scheme, so clearly we have been redirected to a secure address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=141) [Now down in the network traffic, we can see that there was a request made to /Account/Register, which of course](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=149) [was the initial insecure request. That has caused an HTTP 302, which of course is a moved temporarily,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=156) [so a slight semantic difference to what we implemented with web forms, where we wanted it to be moved permanently](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=164) [and then right after that we can see a request for the HTTPS address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=170) [So the implementation in MVC is actually extremely simple because it's just that little annotation on top of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=175) [the controller action. Now that's actually a much nicer way of working because you can go through all those](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=182) [controller actions where you should only ever see them over a secure connection and just add that one-liner](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=187) [on top of the method. Now, of course, you'd implement that on the Login page and the Registration page and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=193) [anywhere else where you might provide information, which you want protected with transport layer security.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=199) [So this is a great way of ensuring that pages can only ever be loaded securely.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=206) [Now, of course, ideally you want to link to the secure scheme because you don't really want browsers to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=211) [making additional requests via redirects. It's always nice to link users to the direct resource that you'd](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=217) [like them to have, but the RequireHttps attribute will step in and make sure that you can never request that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=223) [resource insecurely, even if those other controls, such as linking to secure addresses, are circumvented.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=7&mode=live&start=229)

[Demo: Mixed mode HTTPS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live)

[The next thing I'd like to look at is what we would refer to as mixed mode HTTPS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=0) [So this is where we have a combination of HTTP and HTTPS content on the page and a lot of the time this is not](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=5.5) [very well understood, but I have a demonstration here that should make it quite clear and the first thing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=13.5) [I want to point out is that the page in front of us is loaded over a secure connection, it is an HTTPS address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=18.5) [What I'd like to do now is navigate over to the Contact page. When this page loads, down at the bottom of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=26.5) [the browser we can see a warning -- "Only secure content is displayed".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=33.5) [Now the last thing that you ever want to give users is browser warnings, particularly security browser warnings.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=39.5) [Not only is it a negative browsing experience, it also reflects badly on the site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=47.5) [It makes users ask, why is my browser warning about the security of this website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=53.5) [The other thing is though, because this does tend to happen quite a lot, it starts to desensitize users.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=59.5) [We really only want to see warnings like this when something is wrong and in this case, something is wrong,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=65.5) [and in fact what's wrong is that although this page has been requested over HTTPS, there is another asset](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=72.5) [that is being loaded into this page, which has been requested over HTTP.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=80.5) [Let's open up the Developer Tools and we'll see what that is. So, we'll F12 back into the Tools, start capturing,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=85.5) [and let's just reload the page and you can see the problem down here and, in fact, what we can see here is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=93.5) [that there is a request for an AJAX Library on the HTTP scheme.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=99.5) [So the way this page is constructed, there is an absolute request, which happens to be to another domain,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=104.5) [but we could have the same risk on the same domain if content was loaded insecurely and what we can see here](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=110.5) [is the result was aborted, so Internet Explorer has actually refused to load that resource.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=116.5) [When you think about it, this makes a lot of sense, and the reason is simply this -- showing that a page is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=122.5)[secure by loading it over the HTTPS scheme is intended to give the user confidence in both the confidentiality](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=129.5) [and the integrity of the page. When an external resource, such as a JavaScript file, is loaded in from an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=137.5) [insecure connection, clearly that resource has no integrity guarantee.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=145.5) [Now integrity is important because what it means is, is that a man in the middle could possibly manipulate](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=150.5) [the contents of that JavaScript file. The significance of that is, it makes you ask what could an attacker](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=156.5) [possibly do if they could run whatever JavaScript they like on this page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=164.5) [So even though we're seeing an HTTPS scheme in the address bar, what would happen if we used JavaScript to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=169.5) [say - change the links on the page to go to an attacker's site or maybe embed JavaScript, which could harvest](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=175.5) [any cookies that it could access. So there are actually serious risks with loading insecure content into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=182.5) [a secure page and that's why we got the browser warning; different browsers will present these warnings in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=189.5) [different ways, but certainly most modern browsers do make quite an issue of the fact that there is mixed](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=195.5) [content on the page. Now, of course, the easiest way to avoid this is to make sure that you're never loading](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=201.5) [HTTP content into an HTTPS page, but of course there are still cases where there may be a page which could be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=207.5) [accessed via the HTTP or HTTPS. So do you necessarily want to always load HTTPS?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=215.5) [There's another way of dealing with this and it's called protocol-relative URLs.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=222.5) [Let's jump into Visual Studio and change this reference to the JavaScript file to be protocol-relative.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=228.5) [Now, over in Visual Studio and we're on the Contact view, if we scroll down to the bottom, we will see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=235.5) [where the problem is. So here's what's happening and here is this HTTP address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=241.5) [So, of course, obviously even if this view is loaded over HTTPS, this JavaScript file is going to be requested](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=247.5) [over HTTP and that's where our problem is. The easiest way to solve this is to use a protocol-relative URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=254.5) [and that looks just like this. Now many people will probably look at this and think that something is missing,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=261.5)[where's the scheme? What actually happens though is that when the browser sees an address that starts with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=268.5) [a // like this, it will automatically prefix it with the scheme that the parent page was loaded on.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=275.5) [Let's save this and then we'll jump back over to the browser and see how that works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=282.5) [Right, so back in the browser and we've now got a protocol-relative URL to that JavaScript file.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=288.5) [Let's clear the requests and we'll reload this page. Now the first thing is, is that we don't see a warning](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=294.5) [on the page anymore; there's no more little popup down the bottom of the browser saying that there is a security risk](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=301.5) [and, of course, the reason is simply that that resource is now requested over HTTPS and this time it has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=306.5) [loaded successfully, so that's great. Let's clear this and this time we will request this page over HTTP.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=313.5) [And now we can see the same resource is being requested, except it's being requested over the HTTP scheme.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=323.5) [This is all because of that protocol-relative URL, so this is the one where the script source just starts with //.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=329.5) [This is a great way of requesting external resources because it means you simply cannot get the wrong scheme.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=337.5) [Now, of course, the external resource that you're requesting needs to support both HTTP and HTTPS schemes and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=344.5) [certainly very popular external CDNs of libraries such as jQuery do support both schemes, but there are also](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=352.5) [other external resources that may not support both schemes. So you do need to be a bit conscious of that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=359.5) [Just make sure that both HTTP and HTTPS schemes are supported if you're going to use protocol-relative URLs](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=365.5) [and load the page that embeds them over either secure or insecure connections.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=8&mode=live&start=372.5)

[HTTP strict transport security](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live)

[One thing that we've been talking about very frequently throughout this module is prohibiting the browser from](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=0) [making HTTP requests for resources that we really only want loaded over HTTPS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=6) [Of course, the reason is simply that every single time we make an HTTP request, even if it is redirected to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=14)[HTTPS, that original insecure request is at risk of a man in the middle attack.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=21) [One approach we have to limit the risk of a browser making an HTTP request to a site that we only want to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=29) [HTTPS and certainly there are many instances where we only want a site to serve secure content is to use what's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=36) [referred to as HTTP strict transport security. Now this is commonly abbreviated to HSTS and HSTS is nothing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=43) [more than a response header, which is sent from the server to the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=53) [Now the goal of HSTS is to tell the browser, for this particular site, you should never make an HTTP request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=58) [So even if, for example, HTTP content is embedded on an HTTPS page, which of course would then run the risk](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=66) [we looked at earlier about mixed content or, alternatively, if you never wanted to use it to be able to type in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=74) [an HTTP request and have the browser issue it, even if it does then redirect, the HSTS header is a way of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=80) [ensuring the browser can never make that HTTP request; there is no man in the middle risk if an HSTS header is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=88)[set in the browser. Setting the header is extremely simple and there are a number of different phases throughout](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=97) [the page lifecycle that it can be set. So, for example, BeginRequest is a place where we can set a response header](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=103) [and the header is simply named "Strict-Transport-Security" and it is given a value which is a "max-age".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=110) [Now what the max-age does is it tells the browser any other requests that happened between now and this many](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=119)[seconds into the future must be over HTTPS. So in the example you see in front of you where we've got about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=127) [31 and a half million seconds, that's 12 months. So if a website responds to a request and has an HSTS header](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=135) [like so, the browser will not be able to make an HTTP request to that domain at all within the next year.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=144) [So that's a very nice little security implementation where the site is HTTPS only.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=153) [Clearly this won't work if you still need to be able to make HTTP requests to the site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=159) [Unfortunately, HSTS does have a number of restrictions and the first restriction is simply that it must be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=168) [sent in an HTTPS response. So the browser will not observe that HSTS header if it is sent in an HTTP response.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=175) [Now what this means is, is that in many cases the user still needs to be able to issue an HTTP request,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=186) [which is not interpreted by man in the middle and that's simply because many people never get directly to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=193) [an HTTP address and we'll talk about that more in just a moment.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=199) [So that's the first thing. The other thing is the certificate must be trusted.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=203) [Now that's fine in a production environment, but as we saw earlier on, when you're working in a development](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=208) [environment and you're working with a self-signed certificate, the certificate is probably not going to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=213) [trusted and you're not going to be able to test HSTS. And then, of course, as we just mentioned, you've still](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=219) [got to be able to get that HTTP request through in the first place.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=225) [Unless someone is explicitly going directly to an HTTPS address, which often does not happen, we still have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=230) [a bit of a risk, but the other thing that makes HSTS really rather infeasible, at least in terms of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=238) [a comprehensive mitigation, is that the browser support is very, very patchy.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=245) [So you can use HSTS in Chrome or Firefox and it all worked fine, but if you're using Internet Explorer or Safari,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=251) [it will simply ignore it. Now there's nothing wrong with sending an HSTS response header to Safari or](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=258) [Internet Explorer, the browsers just simply won't do anything with it, so it's not actually going to add any](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=265) [defense to either of those browsers. Of course, that's rather unfortunate because it can be a pretty neat](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=271) [feature that's so simple to implement, it's really just a single line, and it can offer additional security](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=277)[for those HTTPS-only sites. In fact, browsers such as Chrome and Firefox have preloaded HSTS lists and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=283) [the lists are actually a little bit odd and it's very, very short, but they include sites such as PayPal and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=292) [LastPass for password management so that right out of the box, regardless of any website configurations,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=298) [Chrome and Firefox won't be able to make HTTP requests to those preloaded lists.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=305) [So, it's a nice little feature, but it doesn't appear to be very comprehensibly implemented.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=9&mode=live&start=312)

[Other insufficient HTTPS patterns](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live)

[As we've seen at various points through this module, there are quite a number of ways of implementing transport](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=0) [layer security, but doing so in an insufficient manner. So, again, we come back to this concept of -- it's not](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=5) [a matter of having it or not having it, it's a matter of doing it right.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=12.5) [There are a few common anti-patterns that pop up with the way HTTPS is implemented.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=17.5) [One of this anti-patterns we touched on earlier on and that's loading login forms over HTTP and, again,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=23.5) [this goes back to the Tunisia example where the government-run ISPs were injecting JavaScript to harvest](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=29.5) [credentials into the login pages of sites such as Facebook and Google.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=37.5) [Now in those cases, those websites actually posted to HTTPS and there have been many occasions where I've seen](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=42.5) [people say, hey our site is secure; even though we load the login form over HTTP, the credentials themselves](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=49.5) [are protected when the form is posted. Now whilst that's true and the credentials have the advantage of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=57.5) [confidentiality in transit, the login form did not get the advantage of integrity and that integrity allows](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=64.5) [an attacker to harvest those credentials. Another common pattern is you'll see a webpage, which is loaded over](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=73.5) [HTTP, so the address bar will show an HTTP address, and then somewhere on the page there's a login button and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=79.5) [a login form appears in the page. Now for all intents and purposes, it looks like that login form is loaded](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=86.5) [over HTTP. However, what often happens is that that login form is loaded over HTTPS, it's just loaded into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=94.5) [an iframe in the page. It feels like a very integrated experience to the end user, it looks like part of the page,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=103.5) [and naturally people often draw the assumption that it was loaded over HTTP because that's what they see in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=111.5) [address bar. The problem, of course, is that even though the contents of that iframe may come from an HTTPS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=117.5) [request, therefore the login form is strictly HTTPS, the parent page which embeds it is requested over HTTP](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=124.5) [and as a result, we cannot verify the integrity of the parent page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=135.5) [What that means is, is that an attacker could manipulate the parent page so that rather than embedding the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=139.5) [secure login page from the legitimate site, it might embed a login page from the attacker's site and, of course,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=146.5) [because it's in an iframe and you can't see the URL, there is no way of verifying that just by looking at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=153.5) [what you see in the browser. The most secure way of loading that login page is to make sure it is an HTTPS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=158.5) [request and the page is loaded full screen in the browser; you're not embedding it into somewhere else.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=166.5) [Not only does that mitigate that man in the middle risk, but it also means that the end user can see an HTTPS](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=172.5)[address in the address bar. They can also see a little padlock to indicate there's a certificate and they can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=178.5) [inspect that certificate. They are all very important security features and certainly we have been trying](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=185.5) [for many years to educate consumers that they need to see that padlock, they need to see the address bar with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=190.5) [an HTTPS, and depending on the browser, all the sort of visual indicators that go with providing certainty](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=197.5) [that the page is secure. That visibility is extremely important.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=204.5) [The other thing is that for pages that never need to load over HTTP, it should indeed not be possible to load](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=209.5) [them insecurely and we looked at different methods in both web forms and MVC to ensure a resource can only be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=216.5) [loaded over a secure connection. So make sure that pages, such as Logon or Registration pages, cannot be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=223.5) [loaded over HTTP. Even if there are no links to the HTTP page, you don't want someone to be able to go and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=230.5) [change the scheme of the URL and for that insecure page to be loaded.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=237.5) [One final point is that HTTPS does not give complete protection over the contents of the URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=243.5) [So certainly just because a request is over a secure connection doesn't mean that you can start to put](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=251.5) [sensitive information in the address. For example, you don't want to trust credentials in the address because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=257.5) [there are multiple points in the request and response lifecycle where that address may be stored or captured.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=263.5) [So, for example, when a web server receives a request, it will very frequently put that in the logs and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=271.5) [those logs very frequently include the query string. You don't want your user's credentials ending up in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=276.5) [plain text in web server logs. The only real secure place to put that sort of sensitive data is in the request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=282.5) [headers or the request body, definitely not in the requested URL via query string or any other part of the address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=10&mode=live&start=289.5)

[Other HTTPS considerations](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live)

[One thing you do need to keep in mind with SSL is that there is a performance cost.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=0.5) [At the end of the day, we do have the server encrypting and decrypting information and that doesn't come for free;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=6.5) [there must be an overhead. The question, of course, is how much is that overhead?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=13.5) [Is it significant enough that it might warrant additional infrastructure in order to serve more content over](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=18.5) [a secure connection? Fortunately, we do have the benefit of precedence and we have a very good precedent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=25.5) [from Google when they took Gmail to HTTPS-only in 2010. What Google found, and this is a very significant,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=32.5) [very often quoted figure, is that when they went SSL or TLS over all of their requests, it only added less than](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=40.5) [1% of CPU load to the servers. Now on top of that, it also added less than 10-kb of memory per connection and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=49.5) [less than 2% of network overhead, so they are very, very, very small figures.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=58.5) [Now, of course, for an organization like Google who has a huge infrastructure footprint, that's still going to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=63.5) [be a lot of overhead, but when you look at it as a percentage of the available infrastructure, it is very small.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=69.5) [Likewise, the arguments that many people have come up with about the inefficiency of HTTPS, whether that be over](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=77.5) [the network or even around things like caching within the browser, those arguments have pretty much been dispelled](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=84.5) [now; there are really no strong performance arguments against using HTTPS everywhere.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=91.5) [Just implementing HTTPS everywhere, unfortunately, doesn't totally eradicate the risk of a man in the middle attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=100.5) [and I'd like to give you a very good example of why. So let's take a very common scenario here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=107.5) [So the first thing that happens is a user opens up their web browser and they want to go to the American Express](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=113.5) [website so they type americanexpress.com into the browser bar and press Enter.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=118.5) [Now the browser's obviously going to make a request to that website, but the key point here is that it's going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=124.5) [to make that request over the HTTP scheme. That's what the browser will default to if you don't explicitly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=130.5) [put in another scheme, so if you don't explicitly ask for https://americanexpress.com.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=138.5) [Now, when that insecure request is made to American Express, their server will respond with an HTTP 301](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=146.5) ["Moved Permanently" and it will instruct the browser to go and get the website from the HTTPS scheme.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=154.5) [The browser will then go back to the Amex website and request the content over the secure scheme.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=161.5) [So the browser actually has to make an insecure request before it can make a secure request.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=166.5) [The problem, of course, is that that insecure request is at risk of a man in the middle attack and there are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=173.5) [tools such as sslstrip, which can sit with the man in the middle and proxy requests backwards and forwards](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=180.5) [between the insecure request that the user is making and the secure redirect that the server is asking for.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=187.5) [So imagine sslstrip sitting in the middle, we're saving the HTTP request from the user, translating it into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=194.5)[HTTPS, sending it to the server, receiving an HTTPS response, and then turning it back into HTTP and sending it](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=201.5) [to the user, and continually going backwards and forwards like that for all requests, and it's all because of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=208.5) [that one initial insecure request. And, again, that was kind of unavoidable simply because the user typed in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=214.5) [a domain without a scheme. Now, of course, if the user did type in a scheme, that immediately eradicates that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=222.5)[problem and that's a very, very good security practice. Unfortunately, we can't ingrain that in all the users,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=229.5) [but it does solve the problem. Other things that can help solve the problem are HSTS, as we looked at a bit earlier,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=236.5) [but of course that's only limited to Chrome and Firefox. As I also mentioned, Chrome and Firefox come preloaded](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=243.5) [with some HSTS sites, but it's a pretty minimal list so that existing whitelist doesn't really help very much.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=249.5) [There's also a lot of talk about addressing this issue at the DNS level, so what if particular DNS entries](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=257.5) [specify that they could only ever be requested securely. That would be a way of moving the problem from the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=264.5) [web server dictating the secure connection through to the DNS dictating the secure connection.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=269.5)[Of course, that also just moves the attack surface and there are still other DNS-related risks to resolve,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=275.5) [but it does move the responsibility away from the web server. Even if we get all that right though, there is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=281.5) [still the risk of certificate authorities being breached and we've seen that happen on a number of occasions.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=288.5) [Most notably, the Dutch DigiNotar certificate authority in 2011, which was breached and attackers were able](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=294.5) [to create certificates for domains such as Google.com when clearly they had no authorization to do so.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=302.5) [The fundamental fabric of our certificate-based transport layer security is dependent on the integrity of the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=310.5) [certificate authorities. There are quite a lot of them and DigiNotar isn't the only one who's had problems in the past.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=317.5) [Of course, that starts to move away from how we implement transport security within web applications and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=324.5) [really we do have to work on the assumption that the integrity of the certificate authorities is sound.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=11&mode=live&start=330.5)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live)

[Let's summarize the module and probably one of the most important things that should've come through by now](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=0) [is that the risk of insufficient transport layer security is about being insufficient as opposed to being](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=6.5) [on or off, wrong or right. There are shades of gray and we've seen many ways where we can use HTTPS, but](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=13.5) [to do so in a way that doesn't provide a lot of benefit - only posting login forms over HTTPS after loading them](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=21.5) [over HTTP, for example. We saw very graphically how important it is to protect cookies that then give us access](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=28.5) [to other information. Very frequently, the cookie itself isn't considered to be sensitive, but if someone can](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=37.5) [sit in the middle with something like a WiFi Pineapple, get an auth token, and then log themselves in as](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=44.5) [somebody else, that is a serious risk. So cookies need to be flagged as secure to minimize that risk of a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=50.5) [man in the middle attack. Likewise, making sure that resources which require HTTPS can't be loaded over HTTP.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=58.5) [So whether you implement that manually in web forms or whether you use the RequireHttps attribute on an MVC](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=67.5) [controller action, either way it's important to make sure that those resources never have the opportunity to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=75.5) [be loaded insecurely. We also looked at making sure we don't have any of those mixed mode problems, so making](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=81.5) [sure that a secure page can't load an insecure asset within that page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=87.5) [Not only is the browser going to give warnings, but depending on the browser it won't even load the content](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=93.5) [and for very good reason; because of a man in the middle does manage to manipulate something like a piece of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=100.5) [insecure JavaScript, they can change the entire function of a page, even if the page itself was requested securely.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=105.5) [HSTS is a great feature to add for sites that are HTTPS-only, but unfortunately browser support is lacking.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=114.5) [Chrome and Firefox is still significant, but we're going to miss that Internet Explorer and that Safari segment](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=122.5) [of the browser audience and that can be a very significant portion of the audience.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=128.5) [So go ahead and use HSTS, but it is one additional layer on top of everything else.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=132.5) [Remember though security anti-patterns, so things like loading login forms over HTTP or embedding a secure](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=139.5) [login form, but in an HTTP page. So keep in mind that every time an HTTP request is made, not only is there](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=147.5) [a privacy risk in terms of an attacker inspecting it, but there's an integrity risk in terms of an attacker](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=155.5) [manipulating the content. Also make sure that you don't end up with anything sensitive in the URL because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=161.5) [as we discussed, there are multiple points where the URL could be stored and data of a sensitive nature could](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=166.5) [end up persisted insecurely somewhere. And finally, SSL it does have some overhead, it has to.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=174.5) [It's encrypting and decrypting and performing processes that must put some overhead on the infrastructure,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=182.5) [but it's an extremely small percentage - Google believes it's 1 or 2%, that is a negligible figure and certainly](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=188.5) [compared to the value proposition of secure transport layer protection, it's really something that shouldn't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=195.5) [stop people from using SSL. And, of course, the final thing we touched on is even when we do everything right](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=201.5) [with SSL, as of today, there are still risks that make it far from fool proof.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=208.5) [There's still the HTTP to HTTPS redirect that often happens, there's still the risk of certificate authorities](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=214.5) [being hacked and rogue certificates being generated by attackers.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=221.5) [Fortunately, that is a rare occurrence and when it happens, certificates can be revoked and in the case of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=225.5) [a certificate authority such as DigiNotar, they may even cease to exist entirely if they can't protect the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=231.5) [certificates, that's how important that certificate role is.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=237.5) [For us as developers though, we welcome the assumption that the certificate authority itself has integrity](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=241.5) [and we do our best to implement everything else that we've looked at throughout this module.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m9-tls&clip=12&mode=live&start=247.5)

[Unvalidated Redirects and Forwards](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live)

[Introduction](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live)

[Hi, this is Troy Hunt and welcome to the 10th and final part of the OWASP Top 10 on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=0) [Unvalidated Redirects and Forwards, which we also often refer to as Open Redirects.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=5) [In this module, we'll start out as usual by taking a look at how OWASP views the risk of unvalidated redirects](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=14) [and then we will move on to performing an attack against an application, which allows this risk through.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=21) [We'll then talk about what value unvalidated redirects actually pose to an attacker because it's not really](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=28) [a risk on the site itself, but it does pose other problems. We'll have a look at using whitelists and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=35) [referrer checking to thwart malicious use of an open redirect and then we'll finish up by having a look at some](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=41) [other general issues with the risk, particularly around the way different organizations view the severity of it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=47) [So let's move on to taking a look at that OWASP overview and risk rating.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=0&mode=live&start=55)

[OWASP overview and risk rating](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live)

[In the OWASP overview and risk rating, the Threat Agent is considered anyone who might be able to trick users](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=0) [into submitting a request to the website in question and, of course, users can be tricked into following links](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=6) [by a variety of different ways and we'll look at some of those just a little bit later.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=12) [The Attack Vector Exploitability is average and it's really dependent on the attacker providing a link to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=18) [the unvalidated redirect, which then forwards the victim onto another path, normally a malicious path.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=24) [Now this works because the victim is more likely to trust a valid site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=31) [If they see a domain that they recognize and that's the launch point for a malicious attack on another path,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=36) [then the attacker has a much greater chance of circumventing other controls, whether they be human controls](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=43) [such as recognizing a malicious URL, or software controls where malicious URLs may be flagged or prevent a user](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=49) [from following them. So that unvalidated redirect is a very useful tool for adding legitimacy to the attacker's](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=56) [payload. In the Security Weaknesses, OWASP views the Prevalence Uncommon and the Detectability extremely easy](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=63) [and really what this boils down to is applications using a facility to send people to a different address after](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=72) [a particular action is performed. The question is -- can the address that the user is being forwarded to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=79) [be substituted with a malicious address? So, is it actually validated or will the application allow any address](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=86) [to be passed to it, whether it be an internal address, a legitimate external address or a malicious external address.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=92) [What we have to remember with all of those is that when that address is passed through the URL, and we'll see](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=99) [that in practice in a moment, it is again untrusted data and as we've discussed many times in this course,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=105) [untrusted data needs to be validated. The Technical Impact is Moderate and really it boils down to --](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=111) [what is going to happen at the other end of that unvalid redirect?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=121) [So if an attacker is able to pass a malicious path, what is at the end of that path?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=124) [Now, of course, in terms of technical impact, it's really not so much an issue of the vulnerability on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=130) [the website itself, it's rather what does that unvalidated redirect allow an attacker to send a trusting user off to?](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=135) [So it's really using the website as a gateway to something else.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=144) [The Business Impact on this risk is more one of trust. So, what if a user clicks on a link that is your domain](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=152) [and then they have malware installed on their computer. Now, of course, in the case of an unvalidated redirect,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=161)[your domain is simply a gateway. It's allowing the attacker to redirect an unsuspecting user off to that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=167) [malicious path, but it is a trust problem because it does reflect badly on the site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=174) [Somebody clicked on a link to your site and then they got malware. That's the first assumption that they're](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=180) [going to draw so it's more about reputation than actually exploiting data within the site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=1&mode=live&start=186)

[Demo: Anatomy of an attack](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live)

[Let's now take a look at the risk in practice and as usual, I have a sample application to demonstrate it.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=0) [In this case, it is an ASP.NET web forms application and, again as usual, it's just the standard ASP.NET](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=6.5) [template and I've done a little bit of customization. Now what I've done in this case is I've put a few links](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=13.5) [on the front page. So the scenario that I'm trying to recreate here is, you have a website, the website has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=20.5) [some external links, and there's a little process that happens in-between the user clicking on the link and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=27.5) [loading the external site, but first of all let's look at the user experience.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=33.5) [So I'm going to follow this first link to my website and as expected, my website loads.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=37.5) [So, for an end user, that's just a very expected process -- click a link, get the website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=43.5) [Let's go back though because what I want to do is analyze that process with the Internet Explorer Developer Tools.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=49.5) [So, F12 back into Developer Tools, we will go into Network, Start capturing, and let's go down and click on](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=56.5) [that link again and see what happens. Okay, so what we can see here is that the first request is actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=63.5) [to my original site, so that original site was on http://Redirects.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=71.5) [I have a host header entry on my machine so that I can reference that path and it maps through to a local instance of IIS.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=78.5) [So it's very reflective of a normal externally facing website. Now what's important here is that this request](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=84.5) [went to a path called Redirect.aspx and then we can see there's a query string, which passes a parameter called](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=91.5)[URL, and if we scroll that across a little bit, we'll see that that URL passed the address of my website.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=98.5) [So clearly we have got a redirect process, which takes a parameter and then sends us an HTTP 302 to head back out](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=105.5) [to the target site. Now this is very common and it'll often happen for reasons such as logging.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=114.5) [So a site owner says, I'd like to provide links to external sites, but I want to log when my users go there.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=119.5) [That's a very typical example. There are other examples where there may need to be a disclaimer on site exit.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=126.5) [So rather than just an automatic redirect, have a page which says, hey you're leaving the site, we're no longer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=131.5) [responsible for the content. Now that's not as automatic as the example that we see here where the user is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=137.5) [directly sent out to the external path, but it's a very similar risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=143.5) [What I want to do now is show you how an attacker might use this redirect process to trick a user into installing malware.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=148.5) [So let's close those Developer Tools, jump back a spot,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=157.5) [and I have in my Clipboard a path that I'm going to paste in to the address bar and we'll just go back to the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=160.5) [start of that and that's a very similar structure, insofar as we're calling into the redirect process,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=167.5) [we're passing a URL parameter, and then we're passing a URL value.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=173.5) [Now in this case, that URL value is encoded, so it has been encoded for an HTTP address and we can't actually](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=178.5) [see the contents of it, so it's been obfuscated. Let's follow that address and see what happens,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=186.5) [and now we can see down here, we have an Internet Explorer prompt that is asking -- "Do you want to run or save](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=192.5) [malware.exe from evilsite?" So what's actually happened here is that that encoded URL is a link through to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=199.5) [a piece of malware. Now the important thing is, in terms of the risk of unvalidated redirects, is that as a victim,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=207.5) [all we can see in our address bar is a legitimate domain, a domain that we trust.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=215.5) [What's actually happened is that the domain has been used as a launch point to then send the user over to the malware.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=220.5) [This is really the value proposition of an open redirect to an attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=228.5) [They get to socialize a very legitimate, trustworthy-looking domain, but still serve their malware and, of course,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=233.5) [that's valuable because not only does it trick users, but it tricks things like spam filters and it tricks](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=240.5) [other controls that are designed to blacklist malicious URLs because it entirely circumvents it, at least until](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=246.5) [the user clicks the link and then the browser makes a request to that legitimate site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=254.5) [So it's actually a very, very simple attack, but it can be very effective.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=2&mode=live&start=258.5)

[Risk in practice: US government websites](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live)

[There are many precedents of open redirects being used for malicious purposes and one of the most popular uses](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=0) [of an open redirect is for sending spam. A precedent that illustrates this very well is the use of U.S. Government](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=6.5) [websites with unvalidated redirects and, in fact, as we can see here, the article makes a good point of saying,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=15) [would you trust a URL which ends with .gov. So clearly the spam is, in this case, relying on the fact that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=22) [.gov has some sort of implicit assurance to it, it's a very trustworthy address, and that's why it works well](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=29) [for open redirects. When we look at the way this was executed, we can see that the addresses that attackers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=37) [were using were very, very similar to the one that we used just before.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=43) [So there's a URL, which in this case is linkclick.aspx, there's a parameter, which here is just called link,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=47) [and then of course there's the URL. So it's exactly the same execution as what we saw just before.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=54) [What's also interesting here is the sheer numbers that are involved.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=61) [So in this case, we can see that there are over 43,000 clicks redirected to spam domains using the .gov](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=65) [top-level domain name and often these will be through ads, such as you see in front of you here.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=73) [So particularly those work-from-home, part-time, earn lots of money type ads.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=79) [It's advantageous for the spammer to use those open redirects so that first click-through isn't going through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=84) [to a spam target and is picked up as a possible risk by the browser.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=89) [What's also interesting is that this article was written more than half a year ago and the risk in the URL above](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=93) [still exists, it hasn't been fixed, and one of the things we'll talk about during this module is how organizations](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=100) [actually view the severity of this risk because many don't really take it that seriously and sometimes there](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=106) [are good reasons for it, but certainly in cases like this, it appears to just be some sloppy development work.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=3&mode=live&start=112)

[Understanding the value of unvalidated redirects to attackers](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live)

[Let's take a closer look at what the value of an unvalidated redirect is to an attacker and the primary thing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=0) [that we keep coming back to is that an unvalidated redirect gives the attacker the opportunity to exploit trust](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=7) [that the victim has in the target site. Now the victim may be human, so it may be something that they're looking at](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=15) [and they say, hey I recognize that domain, I trust it; the victim may also be an automated tool, such as a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=22) [browser or a spam filter. So a browser or spam filter is not going to flag links on websites or in emails](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=29) [if they are to trusted sites, even though they may then redirect through to malicious sites, and as we saw in](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=37) [the example with the U.S. Government site earlier on, sometimes there are multiple hops through to what is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=44)[ultimately a malicious site and those hops are used to obfuscate the true intent of the end URL.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=49) [In a case such as we see in front of us here, a URL such as this would arouse suspicions.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=56) [Now usually attackers are a little bit more subtle than calling their domain evilsite.com, but certainly there](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=63) [are many tell-tale signs to malicious URLs. It would be unusual, for example, for a link through to an](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=68) [American service to be on a .ru domain. It would also be unusual to see a large company using a very](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=74) [strange subdomain with lots of numbers and letters. This address here is a very different value proposition](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=81)[to this address here. Now obviously trustedsite would be a name or a brand that people recognize and don't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=88) [expect to be malicious and the reality is that when we end up with parameters hanging off the end of the URL,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=95) [people don't pay a whole lot of attention to it. Besides, long URLs normally get truncated in most places](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=102) [where they're displayed so it's very easy for that malicious component of the URL to simply be out of site.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=109) [And then, of course, as we saw earlier on, it's possible to obfuscate the malicious component, so it's possible](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=116) [to URL-encode it all anyway. The main thing is, is that the victim is going to be focused on the domain and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=121) [that's why these two in front of you serve very, very different value propositions.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=129) [Of course, all of this is only in a use if the attacker can socialize this loaded URL, the URL with the malicious](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=136) [parameter hanging off the end of it. How they do that is no different to the way they might socialize many](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=144) [different attack vectors. So, for example, in Part 2, we looked at reflected XSS, which required a victim](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=150) [to click on a carefully formed URL. The distribution method is very similar.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=156) [So they may go out via email fishing scams, they may appear on social media.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=162)[We know that Facebook, for example, is very frequently a point from where scams or attacks are launched.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=167) [And, of course, they may even appear on compromised legitimate sites.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=174) [It's not uncommon for a site, which has an otherwise very honest purpose, to be compromised and for malicious](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=177) [content to be loaded into the site. Now, of course, because this domain is trusted, the whole risk of detection](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=183) [decreases dramatically and this brings us back again to things like spam filters or other native protections](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=190) [built into browsers - those legitimate addresses are extremely valuable to an attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=196) [The other thing is, is that when this address appears in a tool such as Twitter, it gets very, very truncated.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=202) [So normally if somebody does post an address to Twitter, depending on the Twitter client, it's rare to see much](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=209) [more than the domain itself, maybe the first little part of the path.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=215) [Now that's enough to give the victim a sense of what the URL might be, they can look at that domain and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=218) [establish whether they have trust in it or not, but it's not enough to tell the user that there's something](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=224) [malicious going to happen when they do click on that link. And, of course, as I've mentioned a couple of times](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=230) [now, you can always just URL-encode all the parameters and that's what happened in the demo earlier on when](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=234) [we loaded malware.exe. Once parameters are URL-encoded, it is near impossible to visually identify what the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=241) [actual string is. Of course, you can always take that string and URL un-encoded, but certainly most users](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=251) [aren't going to do that and that's what the attacker relies on -- the fact that that path is going to remain obfuscated.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=4&mode=live&start=258)

[Demo: implementing a whitelist](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live)

[One of the things we've discussed a number of times in earlier modules is the concept of untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=0) [and if you cast your mind back, untrusted data is really any information that comes into the system where we](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=6) [can't necessarily verify the integrity or the intent. So certainly data passed in the URL, such as in a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=12) [query string parameter, is untrusted. Now one of the best mechanisms we have for dealing with untrusted data](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=20) [is using a whitelist and a whitelist simply says, this is what we will allow for this data.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=27) [Now whitelists may say that the data needs to adhere to a certain pattern, so maybe it needs to be an integer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=34) [or an email address, but a whitelist can also be very explicit and call out known good values.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=39) [So a whitelist can say, I will only trust this piece of data if it is equal to a or b or c or whatever piece](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=47) [of data we want to use. Now in the case of unvalidated redirects, what we really want to do is be very explicit](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=54) [about the URLs that we will allow the redirect to send a user to.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=62) [We were able to exploit the earlier version simply because an attacker could pass any arbitrary address through](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=67) [to the redirect path and the web application would happily send the user an HTTP 302 and send them off to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=74) [that address. When we look at the code in front of us, you can see why that happens. It's extremely simple.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=81) [We're just going to take that URL from the query string and then send a response redirect to that URL and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=88) [that's what causes the HTTP 302. There's a comment in here, which says "Log exit here".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=93) [Obviously there's got to be a reason for doing a redirect and as I mentioned earlier, a very common reason](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=99) [is that site owners want to log the exit of a person to a link, so maybe they want to log click-throughs to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=105) [partners, for examples. We haven't implemented that just here, but certainly that's the sort of use case](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=111) [that open redirects serve. Now earlier on, I implemented a piece of Entity framework code first to create](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=116)[an entity called TrustedUrl, which as the name suggests simply has a list of URLs that we trust.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=123) [That, of course, then created a persistent storage in the database and I populated that table with the URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=130) [of my website. All of that is really just plumbing work so I won't go through recreating that here,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=137) [but what I will do is now call into it and we'll implement that whitelist.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=142) [So let's start out just by creating a new instance of that data context (Typing) and that's my TrustedUrlContext.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=146) [Now all we need to do is create a simple expression to check if that URL passed into the page via the URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=156) [parameter exists in my collection of trusted URLs. So let's look for db.TrustedUrls and let's just see if](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=163) [there are any (and we'll need to reference System.Linq) TrustedUrls.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=172) [So that is extremely simple and, in fact, what we might do is we might say, if there are not any TrustedUrls,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=182) [then let's throw a new ApplicationException and what we'll do in this case is say, URL not trusted.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=188) [Now normally we probably wouldn't just throw an ApplicationException; we'd handle this a little bit more gracefully,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=198) [but for demonstration purposes, this will give us a really good idea of whether our whitelist is working.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=203) [So let's rebuild that application and then we'll jump back to the web application and see how it works.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=209) [Okay, so back at the web application. Let's first of all reload the page and what we will try next is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=216) [we will try to load the legitimate website. Now I did put http://troyhunt.com into the database of TrustedUrls](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=222) [earlier on, so this should work okay and there we go. So that's loaded.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=231) [So our whitelist works correctly under legitimate circumstances. Let's now go back and this time I'm just](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=236) [going to copy that malicious link again and we'll give this another go.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=243) [So let's paste in the malicious URL, follow that, and here we are -- ‘URL not trusted'.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=248) [So that's an extremely simple implementation, which immediately stops the risk of the unvalidated redirect.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=255)[Of course, the problem is, is that it does create some overhead because now we need to make sure that we have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=262) [a whitelist entry for every single URL leaving the website. So certainly that's something that needs to be](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=267) [factored in and there are use cases where it may not make much sense.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=274) [When you have control over all the instances where you might want to link out to that URL, that works just fine.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=278) [If, for example, there was a situation where users provided their own URLs into the system and you needed to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=285) [have some form of open redirect, that would then be a problem because there's no way to whitelist all of those.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=292) [Your untrusted data moves then to the information that the user provides and submits into the system,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=298) [which others may then later click on, and we'll come back to that because that's one of the areas where](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=304) [unvalidated redirects are sometimes not taken as seriously by providers of services, such as what I've just described.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=5&mode=live&start=310)

[Demo: implementing referrer checking](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live)

[Another very simple mitigation is to look at the referrer header when the user requests the redirect page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=0) [Every time a browser follows a link to a page, when that new page is requested, the path of the page which](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=6.5) [referred the browser is passed in the request header. This is why we're able to do things like look at where](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=14.5) [the incoming links to a website are -- so who has links into my website, for example.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=20.5) [It's a very useful analytics tool. It can also be very useful for trying to establish whether someone has](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=24.5) [legitimately followed a redirect link or whether it's come from an attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=32.5) [So, for example, if we have a redirect such as the one I demonstrated earlier on our sample site here,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=36.5) [you would only expect people to follow a link to the redirect from this site itself.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=43.5) [You wouldn't expect them to arrive at the redirect page from a totally different domain.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=48.5) [So let's have a look at how we can check for that condition. So what I'll do is up before our whitelist,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=54.5) [which we just implemented, I'm going to start out by accessing the referrer property.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=60.5) [So we will create a new variable called referrer and we're going to get that referrer from the request and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=66.5) [on that request object, there is a property called UrlReferrer, which is great.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=73.5) [So that will give us the path of where somebody came from in order to load this page.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=79.5) [Now we can actually start wrapping some conditional logic around this and the first thing I want to do is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=85.5) [check if the referrer is null and simply because if the referrer is null, it means that they did not arrive](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=90.5) [at this page by following a link on another website. Either that or there are some conditions where the referrer](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=97.5) [won't get passed through, for example, if they came from an HTTPS address, the referrer won't be passed with](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=104.5) [the request, but in most cases it probably means that they've, say, clicked on a link in a Twitter client,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=111.5) [which wouldn't pass a referrer, not if it's a rich client on a desktop or alternatively they may receive the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=117.5)[URL in an email. Those sort of circumstances mean that the referrer object will be null.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=123.5) [Now, of course, the other thing that can happen is we could have a referrer and it's just not the same site](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=129.5) [as the one that we're on here. So, we'll add this or condition and what I want to check here is that the host](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=134.5)[of the referrer, or in other words the domain name, if it is not equal to the host of the count page,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=141.5) [so we could look at Request.Url.Host, then we also have a problem.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=148.5) [So either of these conditions will mean that the user has not followed a link on the website through to a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=156.5) [Redirect page. So now we can throw another exception and, again, normally we wouldn't throw an exception like this,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=162.5) [we'd handle it a little bit more gracefully, but this is enough for demonstration purposes.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=168.5) [And let's just say, "Referrer is not the same site" and we'll just rebuild that.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=173.5) [Let's go and have a look now at what happens in the browser. Okay, so back in the browser and, again,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=180.5) [we're expecting that that first link is actually going to work okay.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=187.5) [So let's check this one and that is just fine, that has worked very, very well.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=191.5) [Let's now go back and we will change that Redirect path and, in fact, what we'll do this time, let's just copy](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=196.5)[this address and I will manually change this and we might just make it owasp.org.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=203.5) [Now when we do this, there's not going to be a referrer; we're not following this link from somewhere else,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=210.5) [we're just entering it directly into the browser, and here we have our exception -- "Referrer is not the same site".](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=216.5) [So that has very swiftly circumvented the ability for an attacker to use this by somebody clicking on a link](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=223.5) [in a client that doesn't implement the referrer and, of course, the condition that we put in the code about](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=230.5) [the host of the referrer needing to match the host of our current site that would ensure that even if the link](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=236.5) [is embedded somewhere in another website and the referrer isn't null, it has a value, the host wouldn't match](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=242.5) [the current site either, so we'd get the same exception message.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=247.5) [This is a reasonable mitigation, but it's certainly not without its problems.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=251.5) [For example, if an attacker takes advantage of an XSS floor in a web application, they may be able to circumvent](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=256.5) [the process. The other thing is, is that the referrer header itself could be manipulated by an attacker.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=263.5) [So certainly it's possible for an attacker to manually issue a request and manipulate the request header data,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=269.5) [such as the referrer, but normally that's not what's going to happen when a victim simply follows a link](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=275.5) [in a browser. It's really a different sort of attack vector, but there's also the risk of what would happen](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=281.5) [if the browser didn't send the referrer header. At the end of the day, we're relying on browsers implementing](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=287.5) [that part of the HTTP spec and we all know that sometimes browsers can be a little bit idiosyncratic,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=293.5) [so there is a risk there that things could break for legitimate users if the browser isn't passing that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=298.5) [request header. So certainly it's by no means a perfect or fool proof method, but it can offer some defenses](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=305.5) [under normal operating circumstances.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=6&mode=live&start=312.5)

[Other issues with the unvalidated redirect risk](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live)

[Unvalidated redirects and forwards are always a bit of an anticlimactic way to end the OWASP Top 10,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=0) [simply because they're often viewed as a very "light" risk. So really this is not a vulnerability in the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=6.5) [web application which will expose data or allow an attacker to perform other malicious activity inside the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=13.5) [application itself, rather it's really a launch pad for an attack. It's just the means of an attacker more](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=19.5) [efficiently launching their payload on an unsuspecting victim and, of course, that payload is somewhere](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=26.5) [completely different; it's not on the site in question, but this site has facilitated the process.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=32.5) [One really good indication of the severity of the bug, or rather the lack of severity, is that Google won't](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=39.5) [pay bug bounties for the risk. In fact, they feel that, as they refer to it, are that it's a small number of](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=45.5) [well-implemented and carefully monitored URL redirectors, they don't tend to feel that's a serious risk.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=50.5) [Now, of course, we've got to remember that Google provides services such as Google+ and Gmail, which are](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=57.5) [facilities where users can provide their own URLs into the system and make them accessible to other people.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=63.5) [Assuming Google wants to implement some form of redirect on there, it would really be impossible to implement](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=71.5) [any sort of mitigation, such as a whitelist or referrer checking; things would just simply break.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=76.5) [So their business model really needs open redirects for things to work, but it is a very unique business model.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=82.5) [Certainly the vast majority of websites don't need to have completely open redirects or unvalidated redirects.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=89.5) [Earlier on when we looked at that news article about the U.S. Government websites, it's very unlikely that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=98.5) [their facility needs to be completely open slather. Ideally, they should be whitelisting.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=104.5) [They should simply say, here are the URLs we trust, these are the ones that we know we want to link out to,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=110.5) [and what ultimately is a set of pages curated by staff of the website, so that's a very, very known quantity,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=117.5) [that would be a much safer position for them to be in.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=124.5) [At the end of the day, the real risk to a website is reputation and as we've seen many times now, this is not](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=129.5) [about exposing data on the site or hacking the site; it's about using that site as a starting point for an attack.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=137.5) [The problem, of course, is that from the users' perspective, if they see that address, they're seeing that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=145.5) [legitimate website and that may well end up leading to trust issues or reputation issues.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=152.5) [If your website is continually the one that is ultimately serving up malware, what is the impact of that to](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=157.5) [reputation, because for all intents and purposes, it looks like that site is the one serving up the malware.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=165.5) [The user clicked on a link with your domain name and the next thing they saw was malware.exe.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=172.5) [Now obviously there's another story behind all of that, but certainly the face value doesn't look good for](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=178.5) [the website in question.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=7&mode=live&start=185.5)

[Summary](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live)

[Let's summarize the module, and really this was a very simple risk and, again, it's a very minor risk;](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=0) [it is right down the end of the Top 10. The main thing though is that unvalidated redirects pose value because](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=5) [they establish legitimacy. They exploit the trust that the user has in the website by tricking them into](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=12) [following links that they may not have otherwise done. Now whether it's users or whether it's the fact that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=20) [those open redirects can more easily get through spam filters or more easily get through security controls](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=26) [and browsers, it all comes down to tricking those that need to trust addresses into believing that the URL](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=31) [is legitimate. Now, of course, there are cases where redirects are necessary, but the best mitigation we have](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=38) [is the same mitigation we have for all untrusted data -- implement whitelists, be very explicit about what is](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=46) [allowed, avoid the blacklist approach of saying this is what we don't like because of course that will always](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=52) [be changing; simply say, these are the 10, 15, 100 addresses which we want to allow a redirect to link people](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=58) [back out to. Referrer checking can mitigate the problem, but it won't entirely eradicate it and there are a](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=67) [couple of risks with it as we discussed. It's probably not the best mechanism certainly when there is the](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=73) [ability to implement whitelists. However, if implementing a whitelist was simply going to be too burdensome,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=79) [maybe there's 1,000 links or 2,000 links, referrer checking may be a better option so long as it's not going](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=86) [to break anything for legitimate users. And finally, this really isn't always viewed as a major risk and](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=92) [in reality, certainly compared to something like SQL injection that was right at the start of the Top 10,](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=99) [it's not major. You can't go along to Google, find an open redirect, and then claim your bug bounty, but](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=105) [do consider the risk to reputation. If you are going to implement a redirect and the legitimate sites that](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=111) [it may redirect to is a relatively containable number, just implement a whitelist.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=119) [It's not hard to do and it does address that possible risk of reputation damage.](https://app.pluralsight.com/player?course=owasp-top10-aspdotnet-application-security-risks&author=troy-hunt&name=owasp-top10-m10-redirects&clip=8&mode=live&start=124)