Hello, my name is John Zuccato and I'm a member of the Security Systems Ethical Hacking team. In this video I will be talking about #6 on the OWASP Top 10 List: Sensitive Data Exposure.

What is sensitive data exposure? Well, first of all, what is sensitive data? Normally, this is considered things like banking information, numbers credit card numbers, health information, personal information such as date of birth, your SIN number of Social Security number, or user accounts and passwords.

Sensitive data could be anything your client considers sensitive. What are the implications for this? Well, if somebody steals your account information you could have a financial loss, personal or to your business. Your identity could be hijacked or stolen, and if you are a business, you could have decreased brand trust.

One way that attackers can get access to data is by listening on the wire. Precautions should be used when transmitting sensitive data over the Internet. Data should be encrypted of the transport layer and at the application layer.

If you transfer unencrypted data over the unencrypted channel, anyone listening could get access to the data. In fact, this data breach could happen during transport or possibly at a later date from access to server logs.

So let's go to Altoro Mutual. Here I've uploaded the page. As you can see, this is over HTTP. Now let's go and see what happens when I connect to HTTPS.

I'm running this browser through a proxy, so I'm just going to show you my connections to that briefly. So here I've got a proxy running on my local host on port 9999.

Let's see what happens when I connect over HTTPS.

I get a connection error that tells me this connection is untrusted.

Be careful with HTTPS connections; they're not always trustworthy. When using a secure connection, browsers verify the server identity by checking the validity of the SSL certificate.

There could be honest reasons for an untrusted connection: the certificate authority assigned by an unrecognized authority or self-signed, the certificate has expired, the name of the site and the name of the reported certificate don't match. In general, if you see this message don't trust it.

And definitely don't expose any sensitive data such as your password.

Here's an example of a FireFox warning that communication between Altoro Mutual and the browser cannot be trusted because the certificate is self signed and is for a different domain.

This may mean that we are not connected to the real Altoro Mutual. This could also indicate that the connection is going through a proxy, that someone might be listening in on your Network Communications.

Let's see what happens I login.

Someone's listening in on my traffic and I've logged into the site.

We go in here and I'm going to do a transfer. And it was successful. Awesome!

Let's go see what the traffic proxy tells us. So here's my post request, and I can see my username and password. And then we get logged in. Then I see something interesting here. I see a cookie. And this cookie it's got the user info; the username and the user ID.

Let's see if I could figure out what this is. It's interesting that this password looks like it's encrypted using base 64. Let's see if I can decode it.

Oh, look here. Here's my password: demo 1234. As you can see, base 648 encoding is not encryption. See although the cookie was encrypted, it's easy to determine what the actual value is.

Well, let's see if I can get my banking information as well.

I see my user ID, I've been able to get the username and password, and let's see if I got any other data in here that's valuable.

So here's my transfer. You can see the debit account and the credit card numbers and the amount. So again, we've got all kinds of information here. We've got account numbers, we've got user names and passwords, and my user ID at the bank.

Another way to expose sensitive data is to have insecure cryptographic storage. So another way that attackers can get access to data is through another vulnerability.

In a recent notable case such as the Sony attack, sensitive data, including passwords, has been obtained unencrypted by malicious insider. Sensitive data at rest should always be encrypted to prevent such attacks.

I'm going to show you on Altoro Mutual that there's an SQL injection. Using an SQL injection attack, I'll be able to grab all the user accounts and passwords.

So here I've gone onto the transaction page, and I'm going to do a SQL injection on the "After" field and show you all the usernames and passwords.

Here I have SQL injection attack using the "union" function. Here I'm going to close the first query. I'm going to put a really large date so I won't display any of the values. I just want to show the user names, and I'm going to select the username and password from the user's table.

Let's see what happens when I put that in. So what I've done here is I've appended information from the user table into the current query that displays the transaction ID and account information.

So as you can see here, this is all the user names and corresponding passwords.

Here I have a scan of Altoro Mutual. This is using the general default test policy because there are many tests that will expose sensitive data.

Let's look at the SQL injection ones. So as you see here, we'll look at the transaction and the aspx that shows up in the scan. And it gave me an idea that there is an SQL injection attack on page and I was able to exploit and display the user name and passwords.

So here we have encryption not enforced, and if it's on the login page that's not very good, which means they can go in and login to the site over HTTP.

We have cacheable SSL pages. So these pages are stored locally and if they have sensitive data on them, soomeone can go on view the pages on your PC and get your account information. So here we've got the login page, the transactions page, the transfer page, the account information page.

See, things like email addresses could be considered sensitive data; a lot of times the e-mail address is the user ID or used to describe the user ID.

This missing secure attribute in encrypted session cookie, but when I look at it I see that the user info at cookie is displaying the password and username in a basic 64 encoding so I can easily determine what that is.

Another test that could lead to somebody that steal your password or logging in to a site using your computer is the autocomplete HTML Attribute not Disabled for Password field test. Autocomplete should be disabled for past and feels 'cause this would allow somebody come an open computer and log into a site not even knowing all your passwords.

So as you can see, there are many tests and very tests that will determine sensitive data exposure using AppScan.

Preventing sensitive data exposure encrypt data during transport and at rest.

Use HTTPS to transport data.

Use an up to date security certificate.

Encrypt sensitive data in the database or file system.

Ensure passwords are stored encrypted using a password protection scheme.

Minimize data surface area.

Only store sensitive data when necessary, only transport/show sensitive data when needed.

Use the latest encryption algorithms.

Up to date encryption algorithms should be used and update if known issues are discovered and resolved.

Disable autocomplete on forms that collect data.

Disable caching on forms that collect data.

This concludes our coverage of issue number six on the OWASP Top 10 list. I hope you found this information useful and wish you the best of luck in writing and maintaining secure software