Q16. Following up the previous question, could you please describe us some reasons why you failed to identify security defects during your code reviews?

Theme	Code	Definition	Number of Respondents
Changeset characteristics	Complexity of software	Software architecture is complex	13
	Complex code	Code patch which is under review currently is complex.	5
	Due to obfuscated feature of a programming language	Confusing feature of a programming language	4
	complex patch	Complex pull request	1
	Larger patch	Big pull request	4
Lack of knowledge	Lack of knowledge about software/particular area	Insufficient knowledge about particular project reviewer is working on	16

	Lack of knowledge about security	Insufficient knowledge about potential security issues	8
	security issues related to networking	Insufficient knowledge about security related issues that might occur in the particular project	1
	Lack of experience in code review	Limited prior knowledge about how or where security defects can exist in code	7
	lack of experience in security	Limited history of writing code/reviewing code in part of the project which is vulnerable to security	7
	lack of knowledge about atypical configuration	No prior knowledge about an uncommon configuration which may arise in application	1
	Lack of knowledge about memory management: any sorts of memory issue, use-after-free	Limited knowledge about writing/reviewing memory safe code	5
	lack of understanding about performance characteristics	Limited of knowledge about how to increase performance of application without introducing security issues	1
	Unfamiliarity with the language	No/limited knowledge about the programming language of the project which is under review	1

	False assumption about security	Due to limited/lack of knowledge, any assumption is made which results into a security threat	4
Insufficient checking	not following security guidelines	Not following established guidelines of security for programming language or in overall	1
	Missing check for use-after-free	not checking for use of any pointer after it has been freed	2
	Missing edge case	Any edge/corner case which is missed in unit testing and creates a security hole.	7
	Missing check of third-party package	Not checking properly if the third-party package which is being used in the project, works properly or does not introduce any security threat	2
	Lack of automated testing	No/limited use of automated tools for identifying bugs	2
	missing check for proper synchronization	Insufficient check for the synchronization	3
	inadequate unit test	Unit test has missing cases which may create security issues	7
	Insufficient verification of user input	All possible user inputs were not tested against the feature/application	3

Lack of thorough review	hard to consider all threats/human limitation	As a human, often it is difficult to think of every possible security issue which may arise from the patch under review	20
	Rush reviewing	Not spending enough in code review	9
	Lack of attention	Giving less attention than it is required for code review	26
	Laziness	Acting lazy to dig into security threats while code reviewing	1
	Tiredness	Tired	3
	Overconfidence in other peer	Reviewer is so confident on the developer's skill and experience that he did not check for security threats in code	2
Vulnerability characteristics	Issue triggered from a file outside of patch	The patch which is under review has called something from another file which is not being reviewed currently. An issue triggered from the second file.	7
	Subtle issue	Complex issues that are hard to detect	1
	Novel exploit	Newly generated security issue which was not thought of or seen before	3