**Question 1: When it comes to Smash the Stack, what are the stack canaries? Give a definition and an example.**

Answer: Stack canaries, named for their analogy to a canary in a coal mine, are used to detect a stack buffer overflow before execution of malicious code can occur.

Stack canaries, named for their analogy to a canary in a coal mine, are used to detect a stack buffer overflow before execution of malicious code can occur. This method works by placing a small integer, the value of which is randomly chosen at program start, in memory just before the stack return pointer.

**Question 2: The following code contains errors that could be exploited. How would you fix it?**

***hashOut.data = hashes + SSL\_MD5\_DIGEST\_LEN;***

***hashOut.length = SSL\_SHA1\_DIGEST\_LEN;***

***if ((err = SSLFreeBuffer(&hashCtx)) != 0)***

***goto fail;***

***if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)***

***goto fail;***

***if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)***

***goto fail;***

***if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)***

***goto fail;***

***if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)***

***goto fail;***

***goto fail;***

***if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)***

***goto fail;***

***err = sslRawVerify(...);***

Answer: There were two goto fail; lines in a row, which causes the conditional to always jump to fail, therefore bypassing the signature check, Which creates a security vulnerability.

**Question 3: How could you catch this error in the future?**

Answer: By compiling with address checking and standard memory allocator (hybrid analysis)

**Question 4: Heartbleed bug could be traced to a single line of code. What was that line of code?**

Answer: memcpy(bp, pl, payload);