**Summary of the Articles**

Anonymous is a group of hackivists that have become well known for their cyber-attacks on the cyber infrastructure of corporations and government websites, although they have shown a physical presence in organized protests with the latest being a part of Operation BART and Occupy Wall Street. Methods deployed by Anonymous for cyber-attacks consist not just of distributed denial-of-service attacks (DDoS), but also exploiting “weaknesses in applications, thus allowing them to bypass, at least initially, conventional network defenses such as firewalls and anti-virus applications to access sensitive data” (R1).

Anonymous deployed their methods of cyber-attacks on HBGary and HBGary Federal earlier this year in February after then-CEO Aaron Barr threatened to release the names and information of supposed Anonymous members after Barr supposedly thought he had the upper hand on Anonymous. The threat of releasing information about persons that were supposedly a part of Anonymous didn’t sit well with Anonymous since the information may contain information about innocent people who have nothing to do with the collective. The “punishment” done by Anonymous to HBGary was “swift and humiliating”; swift in that the hacks were done within hours after the threat by Aaron Barr and humiliating because HBGary is a computer security firm that offer services and software to private and public sectors (R2, R3).

While most of the cyber-attacks done by Anonymous are DDoS, such as the PayPal and MasterCard attacks, the plans for HBGary were more sophisticated than just brute force DDoS. The Anonymous members responsible for the hacking, used SQL injection to get information from the databases because of the flawed content management system that HBGary Federal’s website was powered by. Other faults came from poor password security and social engineering.

The first type of attack, SQL injection, can be done because of faulty parameters in the URL when passing parameters between the Web front-end and back-end. “The exact URL used to break into hbgaryfederal.com was http://www.hbgaryfederal.com/pages.php?pageNav=2&page=27” (R3). With SQL injections, an attacker can pass specific relational database commands as parameters to the database to have it execute the queries that the attacker wants done. The URL used to break into hbgaryfederal.com has 2 parameters that have to be handled by the CMS, but was incorrectly handled and allowed for the SQL injection to work which gave access to Anonymous members. In this situation, Anonymous members grabbed the user database from the CMS (R3). Even though they were able to grab the user database that had the list of usernames, emails, and password hashes, the Anonymous members involved still had to crack the hashed passwords.

The hbgaryfederal.com CMS used MD5 for hashing passwords, but didn’t utilize salting and iterative hashing, which increased the vulnerability of all the passwords to become cracked by a rainbow table tool. Along with the poor use of password hash implementations, another flaw with the passwords was the simple combinations used to create them and that would be how Aaron Barr’s password would be cracked with ease (R3). With Barr’s password, Anonymous gained access into all the email accounts for HBGary because Barr used the same password in different places, which also explains why Barr’s Twitter and other social media accounts were hacked and defaced. With access to all of the company’s emails, Anonymous was able to get a hold of useful information from Greg Hoglund, co-founder of HBGary, pertaining to root passwords to the machine running rootkit.com (R2, R3).

Using social engineering, Anonymous members were able to gain root access into the machine running rootkit.com to deface the website. But Anonymous went further than just defacing the website, they also dumped the user database with the emails and password hashes, which also used MD5 the same way the CMS did for hbgaryfederal.com (R3).

After the cyber-attacks, attacks continued, by flooding HBGary offices with phone calls and voicemails, spamming fax machines with Anonymous outpourings, and even leaving a note directed to the company at the RSA security conference.

**How It Relates to the Enterprise Domain**Cyber-attacks on enterprise organizations can be very costly and humiliating such as the one done by Anonymous on HBGary and HBGary Federal. HBGary, being a computer security firm, knew the best practices for securing their websites and machines, but didn’t implement them to deter such attacks like those done by Anonymous. The attacks by Anonymous weren’t complex or “exceptional”, because the hackers “used standard, widely known techniques to break into systems, find as much information as possible, and use that information to compromise further systems” (R3). If HBGary had implemented best practices, then even if the SQL injection were to be a success, Anonymous would not have been able to succeed as much as they had (R3). Best practices are there for reason and that’s to keep the quality of protection high especially if it’s susceptible to cyber-attacks.

Protecting data from hackers at the enterprise domain is a complex and difficult task and that is why “should a cyber attack occur, ensure backup and recovery procedures are in place and enabled” and that IT admins must understand they are never really done securing the data or the network that that data sits on (R2, R3, R4). Failure to protect that data will result in the attackers getting their hands on sensitive data such as emails and passwords like Anonymous had done and then having them released to the world on the internet. The consequences of having sensitive data released into the public sphere can result in monetary loss from lawsuits and if you were the CEO like Aaron Barr was for HBGary, your reputation and position within the company could also be lost too (R3, R5).

Corporate emails containing sensitive information especially about dealings that would have severe civil liability makes for a case that best practices should always be implemented to avoid these types of security breaches, because if communications came to light about anything damaging to the public eye, such as HBGary developing plans to help the U.S. Chamber of Commerce undermine political opponents even if it involved families and children, would come under intense scrutiny and make your company lose clients and/or partners (R4, R5). In HBGary Federal’s case, after the attacks occurred, consequences fell on partners that were grouped with HBGary Federal: Palantir and Berico. Both companies have stated they have discontinued all ties with HBGary Federal because they “do not condone or support any effort that proactively targets American firms, organizations or individuals” (R4). If best practices for security were practiced, HBGary Federal may not have had to put up with so much scrutiny, and Anonymous would not have been able to get the publicity it has and make a mockery out of HBGary.

**How It Relates to the Current Module**

In Module One, we are creating unit tests for methods that return values or take parameters, but also implementing a security policy to help grant the required permissions for the application. With our tests we are implementing input and output checks, but also with the unit tests we are checking to make sure our expected results meet the required results.

The cyber-attacks on HBGary by Anonymous relates to the current module because of our primary focus on the testing and security in Module One. One example how these cyber-attacks would be related to this module is how Anonymous used SQL injection to get into the database to get information. The SQL injection is a form of input, and since we’re testing for input, in this example we would implement best practices for protection against SQL injection by making any query inputs database safe.

Having the security policy in place may also help restrict access against any outside intrusion and may also prevent the deletion of backup data which happened with HBGary’s server once Anonymous hackers were able to get in. Similarly, “you must implement a layered defense of reasonable security controls” which in this module, it seems to be what we are working towards by having input and output checks, a security policy file enforced using the SecurityManager, and other security concepts within our core functionality (R2).