Network Vulnerabilities From A Hacker's Perspective

Terry Yin

October 11, 2014

1 Benefit Of Thinking Like A Hacker

A hacker is a person who solve problem by breaking things, or more commonly, "A hacker is one who enjoys the intellectual challenge of creatively overcoming and circumventing limitations of programming systems and who tries to extend their capabilities." Coleman (2010)

So, a hacker knows how to break things. They know where to look for the unusual places in the system that can help them to achieve their goals. These unusual places are quite often security vulnerabilities of the system. On the other hand, a hacker doesn't follow the rules and dos not limited themselves with any "security models". This means, we may find network vulnerabilities more efficiently by thinking like a hacker.

2 How Hackers Find Vulnerabilities

2.1 The Quantity Makes The Difference

The abstraction in Computer Science is a simplified specification about how something should work. The purpose of having an abstraction is to build other systems based on it without caring the details under it. There are some systems normally work perfectly according to the abstractions. But when the quantities that are handled by these systems exceed certain points, the abstractions will stop working. A hacker is aware and interested in the differences bring by the quantities.

The Distributed Denial-of-Service attack (Wikipedia 2014*b*, DDoS) is one of the examples where quantity makes a difference. A server on the Internet might work fine when all the visits are from limited and intended users. But when there are too many requests, the server might not have enough resource to handle them in time. A DDoS attack is to use many computers on the network to send lots and lots of requests to the victim server so that it becomes unavailable to the intended users.

Buffer Overflow Wikipedia (2014*a*) is another example. When the memory buffer that a computer program reserved for holding a certain input is not big enough, the input data might be written to the adjacent memory. Thus will trigger unexpected behavior from the program, or even the entire system. Hacks know how to use these overflowed data to manipulate the program and eventually take over the control.

2.2 When Quantify Is Not Their Friend, Time Is

Quantity is not always the friend of hackers, but hackers know how to deal with it. Sometimes, quantity is the challenge for a hacker. For example, to guess a password is very hard, because there are too many (not infinite, though) possibilities. A hacker might use a **Brutal Force** approach to crack it. For example, they may use a "dictionary" of common password people often use to try them one by one.

You might think that's not possible because typically a system will deny a user from trying to login after several failed attempts. But hackers can take the encrypted information "home", and crack the password by these encrypted information at “home”. And they will come back only when they have a seemingly right password. So time actually becomes their friend.

2.3 When The Others See The Abstractions They See The Details

Normally, engineers view the system as layers of abstractions. A computer network is a good example. But a hacker won’t think in layers. From their perspective, a computer system is just lots of detailed technologies that together makes things happen.

A hacker might use the **ARP spoofing** (Brookshear 2011, p.176) to send fake data over the low level ARP message over the Local Area Network. Then a flawless service that depends on the high level TCP will be compromised by this attack.

Say, a bored hacker stuck in an airport and couldn’t afdord the Internet fee, but he found the DNS service is available in the wireless network without signing in. The lack of resource might be a problem for a hacker, but the lack of abstraction isn’t! The hacker might build a TCP/IP layer over the DNS service and start surfing the free Internet. (Merlo et al. 2011, DNS Tunneling)

Another example is SQL Injection Wikipedia (2014*e*). When most people see the fields in a Web form as “user name”, “email address”, “phone number”, a hacker might see them just as SQL code snippets.

2.4 Side-effects

Hackers like side-effects and know how to use them.

The **Heartbleed** bug is revealed in April 2014 in the OpenSSL cryptography library Wikipedia (2014*d*). OpenSSL library is widely used in the services on the Internet. This software bug allows more data to be read then needed. So the client will receive some extra “random” data from the server. Because the data area in the server memory is used for keeping user private information, the extra data is not really “random” and might include information that the current user is not granted to access, including the other user’s information. This is a side-effect that will interest a hacker.

A network equipment in a telecom service provider’s computer center might have a USB port. The USB port was designed for uploading and downloading files. But the USB driver in the network equipment could be from some open source software, which doesn’t only do file transfer but can also connect a USB sound card. The file transfer feature might be well tested, but the sound card feature is useless. A hacker can utilize the bugs in the USB sound card driver, and hack the entire network equipment by secretly plugging in a small USB device. Nobody would ever notice.

The above examples showed the vulnerabilities brought by unwanted side-effects. So making a computer system shouldn’t just involve adding wanted features, but also removing unwanted feature. The process of reducing unwanted side-effect is called **hardening** Kopp (1997).

3 Does A Hacker Hack Herself?

The verb or action “hack” in computer security means “to break into computers and computer networks”; in computer science, it means “an inelegant but effective solution to a computing problem” Wikipedia (2014*c*). So when we are talking about creating your own solution, we should take the second meaning.

So, no, a hacker won’t hack herself. Because a hacker solves problem rather than creating more problems and they always seek to become better (Raymond 2003, How To Become A Hacker). A hacker knows the importance of an elegant solution. She has read too much ugly code that’s vulnerable.

4 Conclusion

To create a less vulnerable computer network system, we should not only think like a hacker but also work like a hacker. Hackers do not limit themselves by any models, and network vulnerabilities can be anywhere. Hackers have sharp eyes to see through abstractions, and they don't let side-effects go easily. Hackers don't hack their work.

References

Brookshear, J. G. (2011), *Computer science: an overview*, Paul Muljadi.

Coleman, G. (2010), 'The anthropology of hackers', *The Atlantic* p. 2.

Kopp, C. (1997), 'Hardening your computing assets', *Asia/Pacific Open Systems Review. Computer Magazine Group, NSW under the title of "Information Warfare—Part* **2**.

Merlo, A., Papaleo, G., Veneziano, S. & Aiello, M. (2011), A comparative performance evaluation of dns tunneling tools, *in* 'Computational Intelligence in Security for Information Systems', Springer, pp. 84–91.

Raymond, E. S. (2003), 'How to become a hacker', *Database and Network Journal* **33**(2), 8–9.

Wikipedia (2014*a*), 'Buffer overflow — wikipedia, the free encyclopedia'. [Online; accessed 11-October-2014].
URL: *http://en.wikipedia.org/w/index.php?title=Buffer_overflow&oldid=626899160*

Wikipedia (2014*b*), 'Denial-of-service attack — wikipedia, the free encyclopedia'. [Online; accessed 11-October-2014].
URL: *http://en.wikipedia.org/w/index.php?title=Denial-of-service_attack&oldid=628762198*

Wikipedia (2014*c*), 'Hack — wikipedia, the free encyclopedia'. [Online; accessed 11-October-2014].
URL: *http://en.wikipedia.org/w/index.php?title=Hack&oldid=617604388*

Wikipedia (2014*d*), 'Heartbleed — wikipedia, the free encyclopedia'. [Online; accessed 11-October-2014].
URL: *http://en.wikipedia.org/w/index.php?title=Heartbleed&oldid=629015819*

Wikipedia (2014*e*), 'Sql injection — wikipedia, the free encyclopedia'. [Online; accessed 11-October-2014].
URL: *http://en.wikipedia.org/w/index.php?title=SQL_injection&oldid=627843772*