Alexander Cannell

CSIA 6210

Week 8 - Getting familiar with hackers and hacks

**To get more familiar with hackers, find "The Hacker Manifesto" that was posted in phrack and answer these questions:**

**1) What does being a hacker mean?**

The term Hack originated in the 1940’s with the Model Train Club at MIT. The purpose of the club was to constantly change and modify the functionality and design of model trains. They defined a hacker as someone who applies ingenuity to create a clever result called a Hack.The term hack was used to describe the ability to manage a given challenge. Hacker is a type of person interested in exploration, usually involving computers.

**2) Is a hacker a criminal? Why or why not?**

It depends on the Intent of the hacker. We have several different types of hackers white hat, grey hat, and black hat. A white hat is used to describe a hacker who has permission to be on a given system. Usually these include Pen Testers, Security Analysts, Etc… A Black Hat is used to describe a hacker who has malicious intent and does not have permission to be on the system. Grey hat is the vigilantes of the hacking world. They have a non malicious intent, also no permission to be on a given system with a purpose of curiosity. I would argue if a hackers doesn't have malicious intent then it is not a criminal act. So White Hat and Grey hats are not criminals in my eyes. A lot of companies agree too like Facebook and United Airlines. They actually challenge hackers to find vulnerabilities in their software or websites or systems in general and pays them when they do. These companies view it as a troubleshooting exercise and they are more secure for doing so.

The law would categorize anyone who doesn’t have permission to be a criminal. But I think that is a narrow way of looking at it. I agree with The Hacker Manifesto, “This is our world now... the world of the electron and the switch, the beauty of the baud. We make use of the service already existing without paying for what could be dirt-cheap if it wasn't run by profiteering gluttons, and you call us criminals. We explore... and you call us criminals. We seek after knowledge, and you call us criminals. We exist without skin color, without nationality, without religious bias... and you call us criminals.” These are the new explorers of today. Anyone who has invented something new, created something out of nothing, discovered something new, troubleshooted something, or discovered how something works is a Hacker in their own right.

**3) What is phrack?**

Is an Online Magazine for Hackers and security professionals; where works, news, ideas, and discoveries are published to be tested. The name comes from a combination of the words phreaking and hacking. Also known for The Hacking Manifesto recognized by most hackers.

**4) When did The Hacker Manifesto get released?**

Phracking was posted and written on January 8th 1986. But looks like the very first issue of Phrack was first published on November 17th 1985.

**One of the first tutorials on exploitation was released by Aleph One called "Smashing the Stack for Fun and Profit". Find and read through this article. Answer the following questions**

**5) What are EIP, ESP, and EBP. What's the difference between these three registers**

EIP: extended instruction pointer. The EIP Register contains the address of the next instruction to be executed.

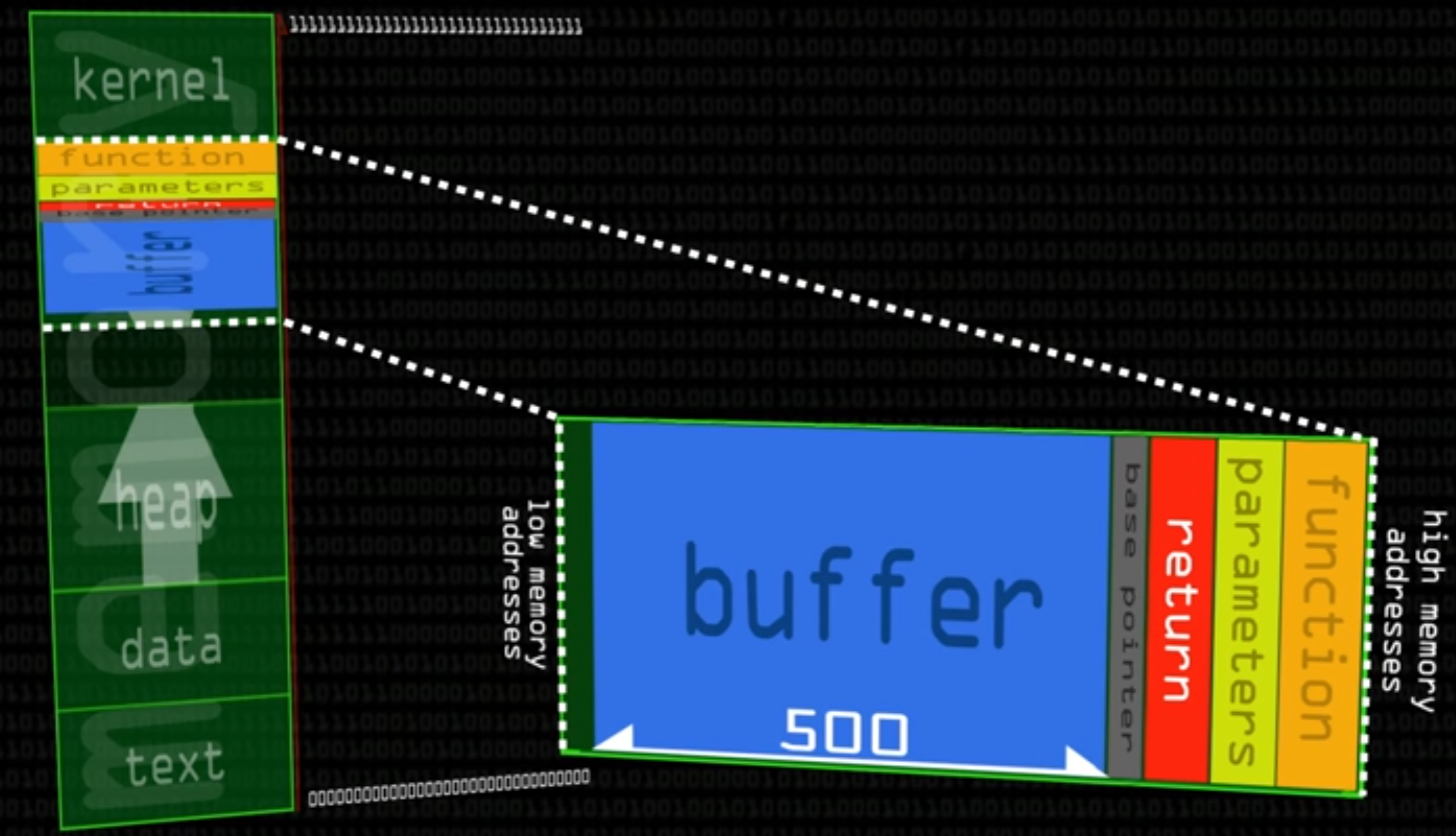
ESP: extended stack pointer. The ESP Register Points to the stacks current location within the stack segment.You can either push or pop the value of ESP. Push stands for decrease, and pop stands for increase.

EBP: extended base pointer. The EBP Register is the base pointer for the current stack frame.

**6) How does a buffer overflow work (2 paragraphs or less)?**

A buffer overflow is a situation where we are using some function to write a string or other variable into some memory that is only a certain length. We are trying to write in something longer than the buffer, it then overwrites the memory addresses. This causes all kinds of problems.

In memory you have Kernel, Text, Data, Heap, and Stack. Kernel is the command line parameters. Text is the actual words in the program, the machine instructions that we compile (read only). Data is the initialized and uninitialized variables are. Heap is where you allocate large things to be stored. Stack is holds the local variables to your functions, and when you call a new function it gets put on the end of a stack. The majority of buffer overflows happen in the stack. Within the stack we have high memory addresses and low memory addresses. The stack is always grows downwards towards the low memory addresses. Within the stack there is something called a calling function. This calling function adds parameters to the stack in reverse order. In the stack we have what is called a buffer which is allotted space. We then have the EBP or the extended base pointer. We have in front of the buffer and the base pointer is the return address. A return address is how main knows where to go after performing the function. The point of buffer overflow is to go over the buffer into the base pointer and get access to the return variable which points back giving us access to stuff.



**7) Research buffer overflow protections. List 3 of them and what they do.**

1. Stack Cookies: The stack or commonly called “canary” which is essentially a random piece of data that the application can be made to write before the EIP. The data overflows from the assigned buffer to the EIP. This overwrites the stack cookies too. This can also be used to check the health of the system. If the cookie has been changed that is a good indication that a buffer overflow has occurred and terminates the application.
2. Data Execution Protection: DEP is available on specific processors, and makes part of the the stack non executable areas. Or in other words any shellcode places in the stack won't execute.
3. Address Space Layout Randomization: ASLR is a feature of many OS’s that randomizes the base address of executables, Dynamic link library (DLL)”s, and others that process in this address space. These addresses change during the system booting process, JMP commands cont be performed in the system DLL’s, and makes their location impossible predict.

**Now let's look at some of the practical examples of what we can do.**

**Go to the site exploit-db.com. This is a database full of exploits for different software. check out what's there and find the exploit-db links for 2 of the vulnerable services on your metasploitable VM.**

**8) What are the two links that you found?**

* Linux/x86-64 - NetCat Reverse Shell Shellcode (72 Bytes): <https://www.exploit-db.com/exploits/41509/>
* MySQL Injection in update, insert and Delete: <https://www.exploit-db.com/docs/41275.pdf>

**9) What do these links do?**

* Linux/x86-64 - NetCat Reverse Shell Shellcode (72 Bytes): A shell is a piece of code or program that can be used to execute a command on a device. A reverse shell is a type of shell that holds open a connection with a listening port on the attacker's computer in order to use code or execute a command on the victim computer. This Exploit uses NetCat to do that. <https://www.exploit-db.com/exploits/41509/>
* MySQL Injection in update, insert and Delete: An SQL injection is when malicious code is embedded into a application and then passed to the backend database. Kindof like a trojan horse. Causing the Database to be un-secure and that embedded code executes commands or queries. <https://www.exploit-db.com/docs/41275.pdf>

**CITED WORKS:**

* "Buffer Overflows." *Buffer Overflows - OWASP*. N.p., n.d. Web. 06 Mar. 2017.<https://www.owasp.org/index.php/Buffer_Overflows>
* "Buffer Overflows 101: What Are They, How Do You Stop Them?" *Enterprise Networking Planet*. N.p., n.d. Web. 06 Mar. 2017. <http://www.enterprisenetworkingplanet.com/netsecur/article.php/3908046/Buffer-Overflows-101-What-Are-They-How-Do-You-Stop-Them.htm>
* Computerphile. "Buffer Overflow Attack - Computerphile." *YouTube*. YouTube, 02 Mar. 2016. Web. 06 Mar. 2017. <https://www.youtube.com/watch?v=1S0aBV-Waeo>
* "ICMP Reverse Shell." *InfoSec Resources*. N.p., 04 Nov. 2014. Web. 06 Mar. 2017. <http://resources.infosecinstitute.com/icmp-reverse-shell/#gref>