IT/CYBR 4323, Chapter 28:

Security

- Slides written by Mr. Donald Privitera to accompany An Introduction to Computer Networks written by Peter L Dordal, http://intronetworks.cs.luc.edu/
- Kennesaw State University
- Marietta Campus
- College of Computing and Software Engineering
- IT Department
- dprivit2@kennesaw.edu

Overview Categories

- Attacks that execute the intruder's code on the target computer
- Attacks that extract data from the target, without code injection
- Eavesdropping on or interfering with computer-to-computer communications

Attacks that execute the intruder's code on the target computer

- The first category is arguably the most serious; this usually amounts to a complete takeover, though occasionally the attacker's code is limited by operating-system privileges. A computer taken over this way is sometimes said to have been "owned".
- Discussed in Dordal book chapter 28.1 Code-Execution Intrusion.
- Perhaps the simplest form of such an attack is through stolen or guessed passwords to a system that offers remote login to command-shell accounts.
- More technical forms of attack may involve a virus, a buffer overflow (28.2 Stack Buffer Overflow and 28.3 Heap Buffer Overflow), a protocol flaw (28.1.2 Christmas Day Attack), or some other software flaw (28.1.1 The Morris Worm).

Code-Execution Intrusion

- The most serious intrusions are usually those in which a vulnerability allows the attacker to run executable code on the target system.
- The classic computer virus is broadly of this form, though usually without a network vulnerability.
- For networked targets, once an attacker is able to run some small initial executable then that program can download additional malware. The target can also be further controlled via the network.
- An executable-code intrusion may be limited by privileges on the target OS, but damage can still be done such as ransomware, and OS vulnerabilities may allow privilege escalation.
- On servers, it is standard practice to run network services with the minimum privileges practical, though see 28.2.3 Defenses Against Buffer Overflows.
- "Executable code" is hard to define. Scripting languages usually qualify. In 2000, the script virus, ILOVEYOU virus began spreading on Windows systems. The .vbs extension, not displayed by default, meant that when the file was opened it was automatically run. The year before, the Melissa virus spread as an emailed Microsoft Word attachment; the executable component was a Word macro.
- Under Windows, a number of configuration-file formats are effectively executable; among these are the program-information-file format .PIF and the screen-saver format .SCR.

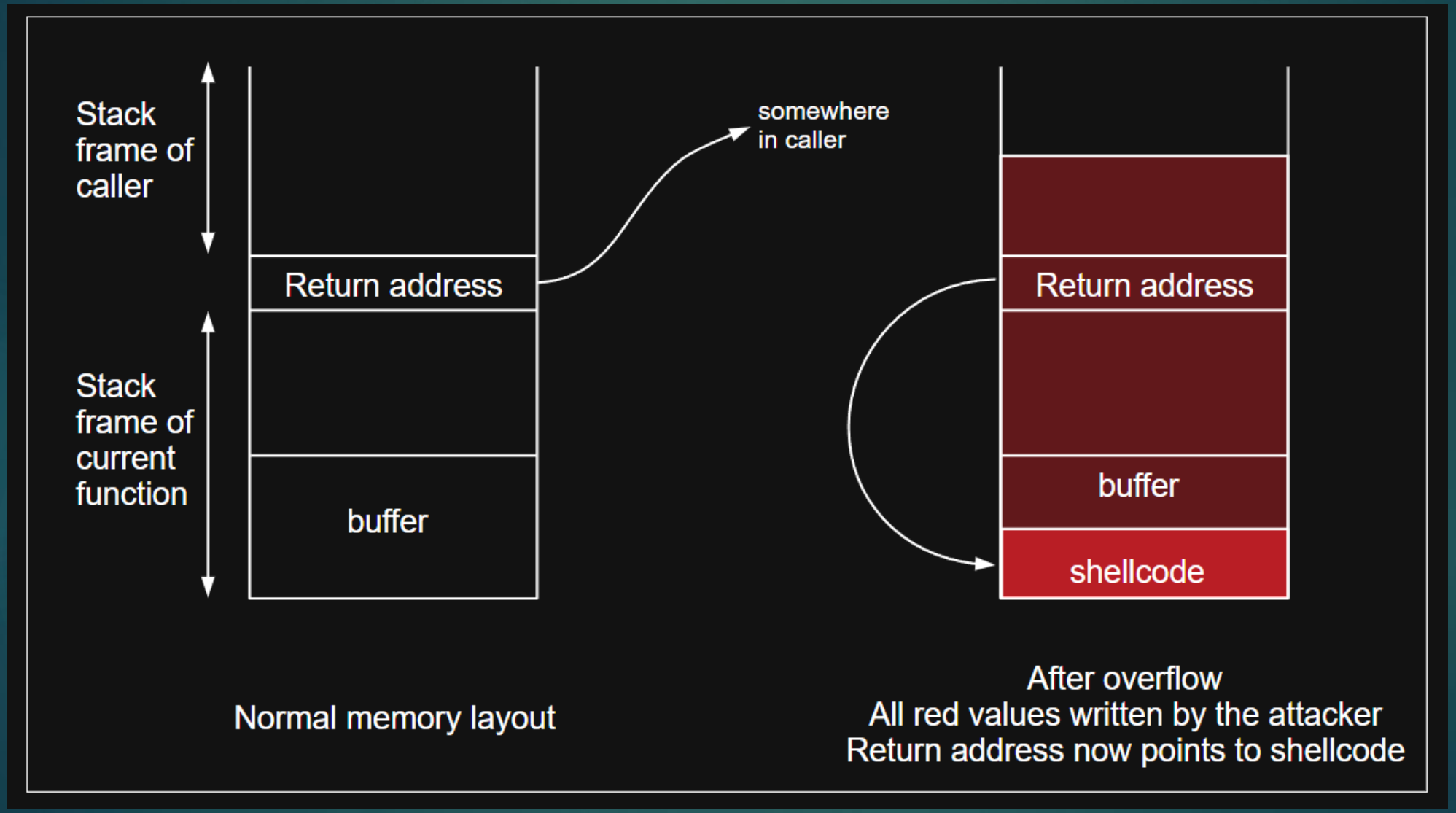
Stack Buffer Overflow

In most memory layouts, the stack grows downwards; that is, a function call creates a new stack frame with a numerically lower address. Array indexing, however, grows upwards: buf[i+1] is at a higher address than buf[i]. As a consequence, overwriting the buffer allows rewriting the most recent return address on the stack. A common goal for the attacker is to supply an overflowing buffer that does two things:

1. it includes a shellcode - a small snippet of machine code that, when executed, does something bad (traditionally but not necessarily by starting a shell with which the attacker can invoke arbitrary commands).
2. it overwrites the stack return address so that, when the current function exits, control is returned not to the caller but to the supplied shellcode.

Peter L Dordal, An Introduction to Computer Networks

Stack Buffer Overflow Diagram



Peter L Dordal, An Introduction to Computer Networks

Defenses Against Buffer Overflows

- Array bounds checking
- Stack canary
- Address Space Layout Randomization (ASLR)
- Memory protection – no execute

Heap Buffer Overflow

- The “heap” is a program’s memory buffer (not related to the stack)
- Dynamically compiled programs load code pages into the heap
- If a dynamic code page is overwritten, it can be used to run malware or jump to a malware routine
- No execute does not work here
- Example is JPEG heap vulnerability
- Cross Site Scripting (XSS)
- SQL Injection

Network Intrusion Detection (NIDS)

- The idea behind network intrusion detection is to monitor one's network for signs of attack. Many newer network intrusion-detection systems (NIDS) also attempt to halt the attack, but the importance of simple monitoring and reporting should not be underestimated. Many attacks (such as password guessing, or buffer overflows in the presence of ASLR) take multiple (thousands or millions) of tries to succeed, and a NIDS can give fair warning.
- Most NIDS detect intrusions based either on traffic anomalies or on pattern-based signatures.
- Stealthy packet manipulation can evade NIDS.
- SNORT, Suricata, & AlienVault are examples of NIDS

Cryptographic Goals

Confidentiality, Integrity, Authentication (CIA)

1. Message confidentiality: eavesdroppers should not be able to read the contents. Confidentiality is addressed through encryption.
2. Message integrity: the recipient should be able to verify that the message was received correctly, even in the face of a determined adversary along the way. Integrity is addressed through secure hashes.
3. Sender authentication: the recipient should be able to verify the identity of the sender. Authentication is addressed through secure hashes and public-key signatures.

Encryption

- Symmetric = Same key used both for encryption and decryption. Lower computational requirements. Extensively used.
- Asymmetric = Two unique keys are required, one for encryption only and one for decryption only. Very computationally intensive. Used sparingly to establish secure communications (along with digital certificates) and then transitioning to symmetric key cryptography.

Secure Hashes

- A hash is a computational function run on data to identify a unique value based on the data.
- Two types of hashes: cryptographically secure and insecure
- Insecure hashes have lower computational requirements making them faster. These functions can sometimes compute the same hash value on 2 different data (collision) making them unsuited for secure purposes. Examples are CRC, MD4, MD5, & SHA1.
- Secure hash functions have no known flaws but advances in computational analysis reveal flaws over time. This creates the ongoing need to make sure insecure hash functions are identified and deprecated. Example is SHA256 and larger bits family.

Public Key Encrypytion

- Also known as asymmetric encryption
- 2 unique and different keys are required: one for encryption and one for decryption.
- Unencrypted message Mu encoded with Key1 K1 yields encrypted message Me. Me can only be unencrypted with Key2 K2.
- Mu x K1 = Me; Me x K2 = Mu
- Mu x K2 = Me; Me x K1 = Mu
- The key used to encrypt and be made public and long as the other key is kept secret.

Peter L Dordal, An Introduction to Computer Networks