# IT3789 Security Cyber Attack & Defence


## L10 – Gaining Access (1)

# WITH KNOWLEDGE COMES RESPONSIBILITY

# Gaining Access

**Exploitation**

**Privilege Escalation**

**Understanding Shellcode**

**Remote & Local Shellcode**

**Sniffing**

**Password Attacks**

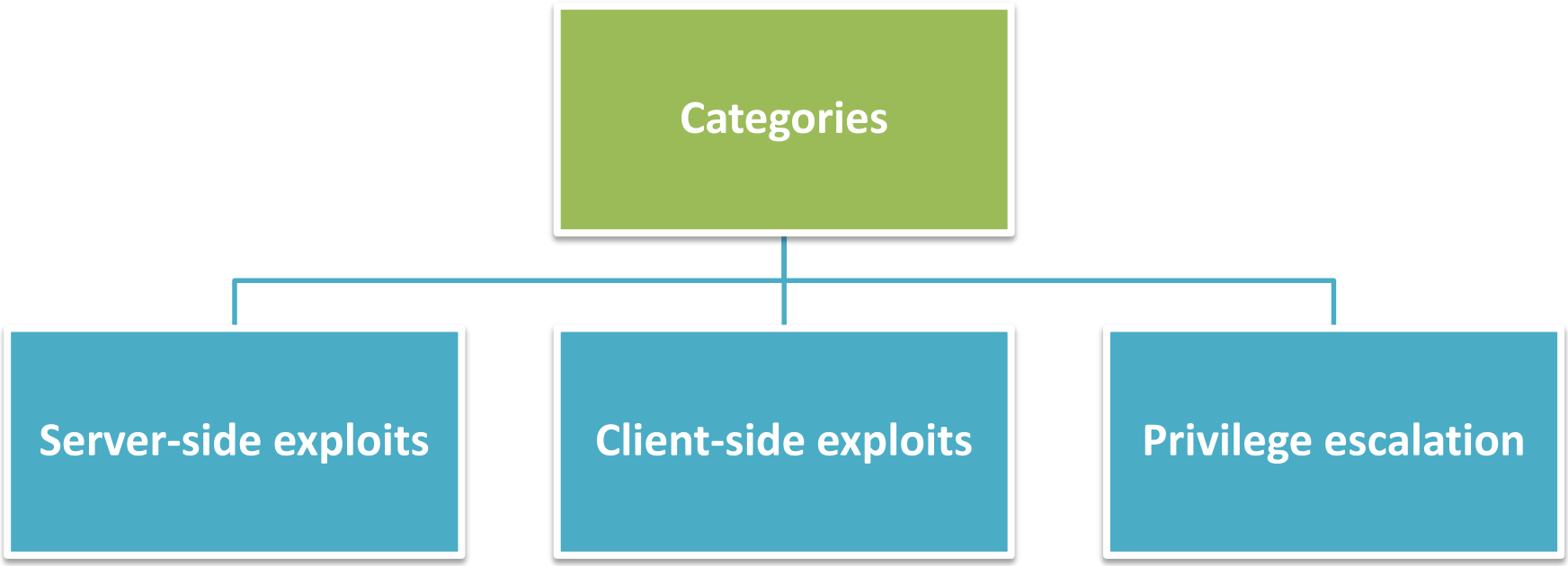
# Exploitation

- Exploit
  - Code or technique that a threat uses to take advantage of a vulnerability.
  - May end up with limited privileges.
- Ultimate aim is to try to gain administrative or root access to the target system.
- A combination of different types of exploits may be needed to gain access to a target system.

# Exploitation

- After exploitation, what's next?
  - Upload programs to compromised machine.
  - Download relevant files from compromised machine.
    - Obtain appropriate information according to the Rules of Engagement.
  - Sniff packets coming in and out of the compromised machine.
    - Get more information about the network and machines interacting with the compromised machine.
  - Reconfigure compromised machine.
    - Use compromised machine to do a pivot attack on other systems.
  - Install software packages in target machine.

# Types of Exploits



# Server-side Exploits

- Vulnerability in listening service.
  - May lead to arbitrary code execution.
  - e.g. Microsoft Security Bulletin MS08-067: SMB Service Vulnerability.
- Attacker attacks service by constructing specially crafted packets to exploit the vulnerability.
- No user interaction is required on the target machine.

# Client-side Exploits

- Client application must access malicious content in order for exploit to work.
  - e.g. Malicious html pages and malicious links in emails.
- Common client applications subject to exploitation.
  - Browsers: Internet Explorer, Mozilla Firefox, Google Chrome.
  - Media players: Quicktime, Realplayer, Winamp.
  - Document: Arcobat Reader, Microsoft Office.
  - Runtime environment: Java JRE.
- Requires user interaction.

# Privilege Escalation

- Prerequisite: Attacker must already have limited privilege on target system to run code.
- Purpose: Attack target system to gain more privilege.
  - Linux/Unix: Gain ***root*** access.
  - Windows: Gain ***administrator*** access.
- May or may not require user interaction.
- Once superuser is attained, the attacker can practically do anything on the target system.

# Risk of Exploitation

- Service crash
- System crash
- System becomes unstable
- System integrity violated
- Data exposure
- Double check Rules of Engagement allows exploitation before proceeding.

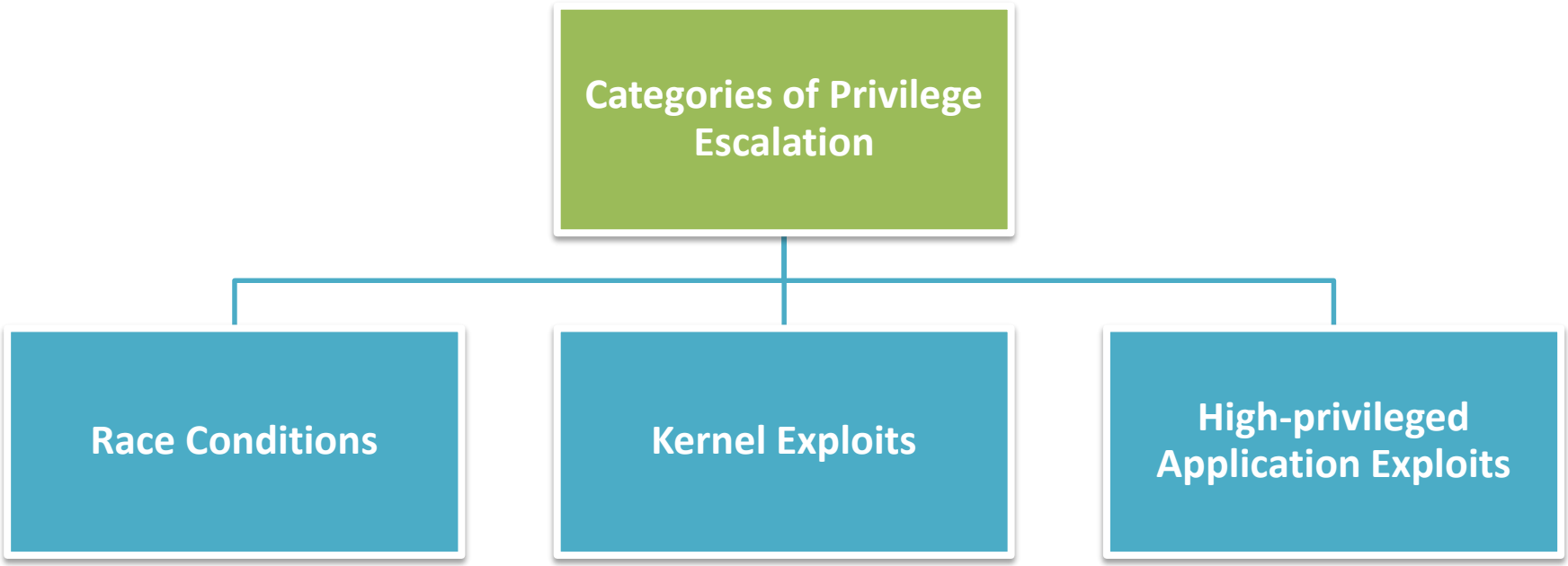
# Gaining Access

| | | |
|---|---|---|
| **Exploitation** | **Privilege Escalation** | **Understanding Shellcode** |
| **Remote & Local Shellcode** | **Sniffing** | **Password Attacks** |

# Privilege Escalation

- Usually it is easier to find a username and password of a non-administrator account.
  - Administrator passwords have more stringent requirements and closely guarded.
- Escalating privilege is the adding of additional rights or permissions to a user account.
  - Makes an account with limited privilege into an administrator account.

# Privilege Escalation



# Privilege Escalation

- Race conditions
  - A flaw that occurs when the timing or ordering of events affects a program's correctness.
  - Involves 2 different actions running on target system in an indeterminate order.
  - Different results will occur depending on sequence of completion of these actions.
  - Some systems have features to check if a program has privilege required to perform a given action.
  - If action finishes before privilege check is done, privilege escalation occurs.

# Privilege Escalation

- Kernel Exploits
  - Flaws within the kernel which allows an attacker to run code that makes calls into kernel functionality.
  - Tricks kernel into running the code with higher privilege.

# Privilege Escalation

- High-privileged Application Exploits
  - Attacks on vulnerable applications that are running with higher privileges such as root or administrative rights.
  - Attacker may be able to trick the application in running code to gain higher privileges.
  - Check high privilege processes running on the target machine.
    - Linux/Unix: ***ps aux*** in shell.
    - Windows: ***tasklist /v*** in command prompt.

# Executing Applications

- Attacker can execute applications after the gaining access to an account with administrator privilege.
- Purpose
  - Install backdoors
  - Keystroke logger
  - Gather sensitive information
  - Damage system

# Gaining Access

| | | |
|---|---|---|
| **Exploitation** | **Privilege Escalation** | **Understanding Shellcode** |
| **Remote & Local Shellcode** | **Sniffing** | **Password Attacks** |

# Assembly Programming Language

- All processor comes with an instruction set that can be used to write executable code for that specific processor type.
  - Assembly source that is written for Intel Pentium processor will not work on Sun Sparc platform.
- Instruction sets are processor type dependent.
- Is small and fast as it is a low-level programming language.

# Shellcode

- A piece of code that is executed when a vulnerability is exploited.
- Usually restricted by size depending on the buffer size available for the vulnerability.
- In hacker's point of view, a shellcode needs to be reliable.
  - If the shellcode is not reliable, target application or system may crash.

# Windows vs Unix Assembly

- Writing shellcode for Windows is different for Unix.
  - In Windows, functions that are exported by libraries are used.
  - In Unix, system calls are used.
- Hardcoding of function addresses in Windows is not recommended.
  - The shellcode may fail if there are changes to system configuration.

# Restricted Characters

- Shellcode cannot contain NULL byte.
- If shellcode contains a NULL byte, it will be interpreted as the string terminator.
  - The rest of the shellcode after the NULL byte are discarded.
  - Shellcode is not executed.
- Need to understand how the program in order to know what characters are restricted.
  - Possible restricted characters for username field for a FTP server
    - Null byte (0x00), carriage return (0x0d), line feed (0x0a) and @ (0x40).

# Tools

- NASM
  - Contains an nasm assembler and disasm disassembler.
  - http://nasm.sourceforge.net
- GDB
  - GNU debugger used to analyze core dump files.
  - Can disassemble functions of compiled codes.
    - Translate C code to assembly language.
  - http://www.gnu.org/software/gdb/
- ObjDump
  - Disassemble files and obtain important information.
  - http://www.gnu.org/software/binutils

# Gaining Access (1)

## Exploitation

- Overview
- Types of Exploits
- Risk of Exploitation

## Privilege Escalation

- Categories of Privilege Escalation
- What attacker can do after privilege is escalated?

## Understanding Shellcode

- What is a shellcode?
- NULL byte problem