- **Course Name :** CodeNext – Malware Development Intermediate Program CN102
- **Course Duration :** 90 Days
- **Course Fee :** INR 20,000/participant

- ✓ **Platform:** Windows 7, Linux
- ✓ **Tools (Windows):** Microsoft Visual Studio 2013, Debugging Tools for Windows
- ✓ **Tools (Linux):** GNU C Compiler, NASM Assembler, GNU C++ Compiler, GNU Debugger, Any suitable C++ IDE for Linux
- ✓ **Pre-requisites:** CodeNext-Malware Development Beginner Program CN101
- ✓ **Project Work:** Projects will be developed by a group of 3-4 people during the course. Projects will range from pure theoretical work to development of a proof of concept malware technique(s).
- ✓ **Course Module:**

1. **Hiding Runtime Traces**
   a. Why is it important to hide traces?
   b. Extension Spoofing: Cheating The Eyes
   c. Stealth Mode
      i. Common Techniques
         1. Processes with Common Names
         2. DLL Injection
            a. Hiding Loaded DLLs
         3. Code Injection
   d. **Practical Stuff**: Hiding a malware (a malware from previous modules)
2. **Hindering with Malware Analysis**
   a. Common Malware Analysis Methods
      i. Static Analysis
         1. Decompilation
         2. Disassembly

Corporate Office: 3rd Floor, 26 Pusa Road, Adjoint Karol Bagh Metro Station Gate No. 4, New Delhi, India
Email: rahul@ignitetechnologies.in **Contact No.:** +91-9599387842,
Website: www.ignitetechnologies.in www.rmar.in

1. Process Monitoring
2. Debugging

iii. Evading from Static Analysis
1. Encrypted Code
2. Anti-Reverse Engineering Methods

iv. Evading Dynamic Analysis
1. Debugger Detection Methods
a. Using IsDebuggerPresent()
b. Using Debugger Flags
c. Using Native APIs
d. Using Timing Analysis
2. Self-Debugging Codes
3. Self-Modifying Codes

v. **Practical Stuff:** Writing malware with anti-malware analysis methods

3. **Cryptomalware**

a. One way malware
i. Using RSA Encryption

b. Safe Data Stealing
i. Using Public Key Cryptography

c. Cryptocomputing
i. Homomorphic Cryptology
ii. Computing using Homomorphic Cryptography

d. Cryptographic Backdoors
i. Significance of random numbers
ii. Generating random numbers
1. Pseudo random number generator
2. True random number generators
3. Cryptographically secure random number generators
a. Blum Blum Shub

   b. Blum Micali
   c. AES-CTR
iii. **Practical Stuff:** Developing backdoor for crypto-systems
   1. Backdoor for RSA
   2. Backdoor for Zero Knowledge Proofs
   3. Backdoor for Secret Sharing Protocols
iv. **Practical Stuff:** Steal data like pro (unprovable data theft)

4. **Network Programming**
   a. Introduction to Computer Networks
   b. Introduction to QT Framework
   c. Network Programming Using QT
      i. Writing "Hello, Network"
   d. Client and Server Model
      i. Creating client and server
   e. **Practical Stuff:** Creating chat server and client.
5. **Remote Access Toolkits**
   a. What are RATs?
   b. Command and control server
   c. Features of a simple RAT
      i. Remote command and control
      ii. Reverse Connection
   d. Creating a simple RAT
      i. Remote Command Execution
      ii. Data Stealer
   e. Creating Command and Control Server
   f. Deniable Communication of RAT
   g. **Practical Stuff:** Creating a basic RAT for Linux server