Phpsploit-Framework Software Technology White Paper

1. Software Introduction

Phpsploit-Framework is an open source CTF framework and vulnerability exploitation development library. It is written in PHP language and designed specifically for penetration testing and security audit behavior, aiming to help ethical hackers (penetration testing engineers, IT auditors, security research and development engineers, etc.) more efficiently and quickly carry out vulnerability mining and security audit work.

The software project address is https://github.com/huc0day/phpsploit-framework Use github as the repository for open source code.

2. Software author

The author of Phpsploit-Framework software (net name "huc0day") is an ethical hacker dedicated to the open source industry. I have served as a research and development engineer, software architect, technical director, information security officer, senior IT auditor, and other positions in multiple internet companies, specializing in related work in fields such as penetration testing, reverse engineering, web security, binary security, and security software development. Formerly a member of the early Red Guest Alliance, he has designed original works such as network security programming framework, memory based database core engine, GDSC data security communication protocol, and server intrusion alarm system since 2004. Contact email: huc0day@outlook.com .

3. Software usage

3.1. The running environment of the Phpsploit-Framework software (web or Cli environment of php5.6+)

3.1.1. PHP language version (5.6+)

3.1.2. PHP extensions: bcmath, openssl, pcntl, pdo_ MySQL, shmop, curl, etc(Note: When some extensions are not installed, some features of the Phpsploit-Framework software will not be available)

3.1.3. Operating system environment: Linux series

3.1.4. Web environment or command-line environment: Apache2, nginx, cli, etc(Note: Some functions of the Phpsploit-Framework software can only run in root and cli mode)

3.1.5. Database environment: Mysql or Mariadb

3.1.6. Browser environment: Most browsers other than IE (such as Google Chrome, Firefox, etc.)

3.2. Running environment for instant messaging client of Phpsploit-Framework software (openjdk-21)

3.2.1. JAVA language version (openjdk-21)

3.2.2. JAVA third-party package: fastjson2-2.0.6.jar、jedis-4.2.3.jar、javacv-1.4.1.jar

3.2.3. Operating system environment: Linux series

3.3. There are three operating modes (development mode, testing mode, and production mode) for the phpsploit framework software:

3.3.1. Development mode: In this mode, including a complete framework file, you can easily expand, optimize, and improve on the existing framework content to make it more suitable for your current scene needs!

3.3.2. Test mode:In this mode, the complete framework files are also included, but the difference is that the software will be run using the relevant configurations in formal mode. In this mode, you can easily debug the software operation and effectively repair any bugs found during the software operation process!

3.3.3. Production mode: In this mode, the framework is packaged as a single project program file without any other dependent files (the file name varies depending on the version type of the project program file (Lite and Full), and the project program file name format is build_< Lite/full>_< 18 digit long integer numbers>. phpsploit.php.The project program files under this mode are widely used for penetration testing and security auditing under the premise of obtaining legal authorization.

3.4. Introduction to the relevant functions of Phpsploit-Framework software

3.4.1. Introduction to Key Function Routing

3.4.1.1. /home //Home directory

3.4.1.2. /user //Account Directory

3.4.1.3. /user_info //Account information

3.4.1.4. /create_production_account //Create privileged account information

3.4.1.5. /unsubscribe //Cancel the account information currently logged in

3.4.1.6. /logout //Exit current login

3.4.1.7. /server //Server Environment Directory

3.4.1.8. /server_info //Display server environment information

3.4.1.9. /session //Server Session Environment Directory

3.4.1.10. /session_info //Display server session information

3.4.1.11. /cookie //Client Session Environment Directory

3.4.1.12. /cookie_info //Display client session information

3.4.1.13. /build //Build project files

3.4.1.14. /encode_build //Building encrypted files

3.4.1.15. /decode_build //Building Decrypted Files

3.4.1.16. /debug //Switch debugging mode

3.4.1.17. /guide //reference list

3.4.1.18. /guide/common_commands //Common Command Introduction Module

3.4.1.19. /guide/user_commands //Account related command introduction module

3.4.1.20. /guide/elf_commands //ELF format content operation related command introduction module

3.4.1.21. /guide/system_commands //Introduction module for operating system related commands

3.4.1.22. /guide/file_commands //Introduction module for file operation related commands

3.4.1.23. /guide/hardware_commands //Introduction module for hardware operation related commands

3.4.1.24. /guide/software_commands //Introduction module for software operation related commands

3.4.1.25. /guide/network_commands //Introduction module for network operation related commands

3.4.1.26. /guide/firewall_commands //Introduction module for firewall operation related commands

3.4.1.27. /guide/webserver_commands //Introduction module for web server operation related commands

3.4.1.28. /guide/docker_commands //Docker container operation related command introduction module

3.4.1.29. /guide/penetration_test_commands //Introduction module for commands related to penetration testing

3.4.1.30. /guide/penetration_test_commands/information_gathering //Introduction module for information collection related commands in penetration testing

3.4.1.31. /guide/penetration_test_commands/vulnerability_analysis //Introduction module for vulnerability analysis related commands in penetration testing

3.4.1.32. /guide/penetration_test_commands/web_program //Introduction module for web program related commands in penetration testing

3.4.1.33. /guide/penetration_test_commands/database_evaluation //Introduction module of database evaluation related commands for penetration testing

3.4.1.34. /guide/penetration_test_commands/password_attack //Introduction module for password attack related commands in penetration testing

3.4.1.35. /guide/penetration_test_commands/wireless_attacks //Introduction module for wireless attack related commands in penetration testing

3.4.1.36. /guide/penetration_test_commands/reverse_engineering //Introduction module of reverse engineering related commands for penetration testing

3.4.1.37. /guide/penetration_test_commands/vulnerability_exploitation //Introduction module for vulnerability exploitation related commands in penetration testing

3.4.1.38. /guide/penetration_test_commands/sniff_deception //Introduction module for sniffing/spoofing related commands in penetration testing

3.4.1.39. /guide/penetration_test_commands/permission_maintenance //Introduction module for commands related to permission maintenance in penetration testing

3.4.1.40. /guide/penetration_test_commands/data_forensics //Introduction module for digital forensics related commands in penetration testing

3.4.1.41. /guide/penetration_test_commands/reporting //Introduction module for commands related to penetration testing reporting tools

3.4.1.42. /guide/penetration_test_commands/social_engineering //Introduction module of social engineering related commands for penetration testing

3.4.1.43. /security //Encryption and decryption function module

3.4.1.44. /security/url //URL encryption and decryption module for encryption and decryption functions

3.4.1.45. /security/base64 //Base64 encryption and decryption module for encryption and decryption functions

3.4.1.46. /security/sha1 //Sha1 encryption and decryption module for encryption and decryption functions

3.4.1.47. /security/md5 //MD5 encryption and decryption module of encryption and decryption function

3.4.1.48. /security/crc32 //CRC32 encryption and decryption module of encryption and decryption function

3.4.1.49. /security/crypt //Crypt encryption and decryption module of encryption and decryption function

3.4.1.50. /security/openssl //OpenSSL encryption and decryption module of encryption and decryption function

3.4.1.51. /security/hash //Hash encryption and decryption module of encryption and decryption function

3.4.1.52. /security/password_hash //Password of encryption and decryption function_ Hash encryption and decryption module

3.4.1.53. /security/sodium //Sodium encryption and decryption module of encryption and decryption function

3.4.1.54. /security/hash_hmac //Hash of encryption and decryption function_ HMAC encryption and decryption module

3.4.1.55. /memory //Shared memory management module

3.4.1.56. /memory/system //Shared memory management module

3.4.1.57. /memory/search //Shared memory resource search module for shared memory management

3.4.1.58. /memory/list //Shared memory resource list module for shared memory management

3.4.1.59. /memory/add //Shared memory resource addition module for shared memory management

3.4.1.60. /memory/clear //Shared memory resource cleanup module for shared memory management

3.4.1.61. /database //Database management module

3.4.1.62. /database/query //Data query module of database management

3.4.1.63. /database/exec //Data update module of database management

3.4.1.64. /file //File Manager Module

3.4.1.65. /file/search //File search module of file management

3.4.1.66. /file/explorer //File browsing module of file management

3.4.1.67. /file/create //File creation module of file management

3.4.1.68. /file/upload //File upload module of file management

3.4.1.69. /file/clear //File cleaning module of file management

3.4.1.70. /scan //Scan management module

3.4.1.71. /scan/webs //Scan management web site survival detection module

3.4.1.72. /scan/domain //Scan management host port opening detection module

3.4.1.73. /scan/tamperproof //Abnormal sample detection and early warning module of scanning management

3.4.1.74. /wget //File online download module

3.4.1.75. /elf //Elf format content parsing module

3.4.1.76. /elf/elf64 //Elf64 format file parsing module for ELF format content parsing

3.4.1.77. /elf/elf_h //Elf format content parsing c language source code definition reading module

3.4.1.78. /shell //Shell environment management module

3.4.1.79. /shell/web_shell //Webshell environment module of shell environment management

3.4.1.80. /shell/server_shell //Server module of shell environment management based on C / S architecture

3.4.1.81. /shell/server_shell_client //Client module of shell environment management based on C / S architecture

3.4.1.82. /shell/reverse_shell //Shell environment management rebound shell environment module

3.4.1.83. /shell/background_shell //Shell environment management rebound shell environment module

3.4.1.84. /shell/proxy_shell //Proxy shell environment module of shell environment management

3.4.1.85. /shell/proxy_shell/create_session_id //Creation authentication information of proxy shell environment

3.4.1.86. /shell/proxy_shell/clear_session_id //Clean up authentication information of proxy shell environment

3.4.1.87. /shell/proxy_shell/send //Send shell command of proxy shell environment

3.4.1.88. /shell/proxy_shell/receive //Receive shell command execution results of proxy shell environment

3.4.1.89. /shell/proxy_shell/listen //Monitor and execute shell commands in a proxy shell environment

3.4.1.90. /chat //Instant messaging management module

3.4.1.91. /chat/server_chat //Instant messaging server module based on C / S architecture for instant messaging management

3.4.1.92. /chat/reverse_chat //Client module of instant messaging management based on rebound connection technology

3.4.1.93. /report //Report management module

3.4.1.94. /report/create_vulnerability_report //Report creation module of report management

3.4.1.95. /report/edit_vulnerability_report //Editing report module of report management

3.4.1.96. /report/show_vulnerability_report //View report module of report management

3.4.1.97. /report/export_vulnerability_report //Export report module of report management

3.4.1.98. /report/clear_vulnerability_report //Clean up report module of report management

3.4.1.99. /clear //Reset the login status and relevant environmental information of Phpsploit-Framework software

3.4.1.100. /logout //Log Out

3.4.2. introduction to key functional interfaces

3.4.2.1. Interface_Root //Basic interface of framework (framework root interface)

3.4.2.2. Interface_Base //The basic function interface of the framework is the library root interface (inherited from interface_root)

3.4.2.3. Interface_Operate //Module function interface library root interface of the framework (inherited from interface_root)

3.4.2.4. Interface_Controller //Controller interface library root interface of the framework (inherited from interface_root)

3.4.2.5. Interface_View //The view of the framework represents the layer interface library root interface (inherited from interface_root)

3.4.2.6. Interface_Main //Program call interface of the framework (inherited from interface_root)

3.4.2.7. Interface_Base_Block //basic interface of shared memory management of the framework (inherited from the framework interface_base)

3.4.2.8. Interface_Base_Block_Data //data interface of shared memory management of the framework (inherited from the framework interface_base)

3.4.2.9. Interface_Base_Block_Indexes //Index list interface for shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.10. Interface_Base_Block_IndexesItem //Index list sub item interface of shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.11. Interface_Base_Block_Keys //The core index interface of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.12. Interface_Base_Block_UniqueIndex //Unique index interface for shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.13. Interface_Base_BlockContent //The block content interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.14. Interface_Base_BlockContentType //The block content type interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.15. Interface_Base_BlockEndFlag //The end of block flag interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.16. Interface_Base_BlockHead //The block header interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.17. Interface_Base_BlockHeadEndFlag //The end of block header flag interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.18. Interface_Base_BlockKey //The block index field interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.19. Interface_Base_BlockMode //The block mode field interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.20. Interface_Base_BlockName //The block name field interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.21. Interface_Base_BlockReserved //The block reserved field interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.22. Interface_Base_BlockSize //The block size field interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.23. Interface_Base_BlockStatus //The block status field interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.24. Interface_Base_BlockType //The block type segment interface in the memory block data structure of the shared memory management of the framework (inherited from the framework interface interface_base)

3.4.2.25. Interface_Base_Document //The basic interface of the document processing function of the framework (inherited from the framework interface interface_base)

3.4.2.26. Interface_Base_Error //The basic interface of the error handling function of the framework (inherited from the framework interface interface_base)

3.4.2.27. Interface_Base_Exception //The basic interface of the exception handling function of the framework (inherited from the framework interface interface_base)

3.4.2.28. Interface_Base_File //The basic interface of the file processing function of the framework (inherited from the framework interface interface_base)

3.4.2.29. Interface_Base_Format //The basic interface of the format conversion function of the framework (inherited from the framework interface interface_base)

3.4.2.30. Interface_Base_FormatType //The format type of the framework defines the basic interface (inherited from the framework interface interface_base)

3.4.2.31. Interface_Base_Lock //The basic interface of the concurrent locking function of the framework (inherited from the framework interface interface_base)

3.4.2.32. Interface_Base_Memory //Shared memory function basic interface of the framework (inherited from the framework interface interface_base)

3.4.2.33. Interface_Base_RawSocket //Network raw socket function basic interface of the framework (inherited from the framework interface interface_base)

3.4.2.34. Interface_Base_Request //Network request function basic interface of the framework (inherited from the framework interface interface_base)

3.4.2.35. Interface_Base_ResourceType //The resource type of the framework defines the basic interface (inherited from the framework interface interface_base)

3.4.2.36. Interface_Base_Response //The basic interface of the network response function of the framework (inherited from the framework interface interface_base)

3.4.2.37. Interface_Base_Security //Basic interface of security function of framework (inherited from interface_base of framework)

3.4.2.38. Interface_Base_Security_Rsa //The basic interface of the asymmetric encryption function of the framework (inherited from the framework interface interface_base)

3.4.2.39. Interface_Base_Shell //Shell environment function basic interface of the framework (inherited from the framework interface interface_base)

3.4.2.40. Interface_Base_Socket //Network socket function basic interface of the framework (inherited from the framework interface interface_base)

3.4.2.41. Interface_Controller_ChatServer //Basic interface of the chat server function controller of the framework (inherited from the framework interface interface_controller)

3.4.2.42. Interface_Operate_ChatServer //The basic interface of the chat server function module of the framework (inherited from the framework interface interface_operate)

3.4.2.43. Interface_Operate_User //User function module basic interface of the framework (inherited from the framework interface interface_operate)

3.4.3. Introduction to Key Functional Classes

3.4.3.1. Class_Root //The basic root class of the framework(Inherited from the PHP language's built-in base class: StdClass)

3.4.3.2. Class_Base //The basic functional class library root class of the framework(Inherit from framework root class: Class_ Root)

3.4.3.3. Class_Operate //Module Function Class Library Root Class of the Framework(Inherit from framework root class: Class_ Root)

3.4.3.4. Class_Controller //The controller class library root class of the framework(Inherit from framework root class: Class_ Root)

3.4.3.5. Class_VIew //View representation layer of framework, library root class(Inherit from framework root class: Class_ Root)

3.4.3.6. Class_Main //Program Call Entry Class for Framework(Inherit from framework root class: Class_ Root)

3.4.3.7. Class_Base_Auth //Login verification class for framework(Inherit from framework base class Class_ Base)

3.4.3.8. Class_Base_Block //Basic classes for shared memory management of frameworks(Inherit from framework base class Class_ Base)

3.4.3.9. Class_Base_Block_Data //The shared memory management data class of the framework(Inherit from framework base class Class_ Base)

3.4.3.10. Class_Base_Block_Indexes //Index List Class for Shared Memory Management of the Framework(Inherit from framework base class Class_ Base)

3.4.3.11. Class_Base_Block_IndexesItem //Index List Subitem Class for Shared Memory Management of the Framework(Inherit from framework base class Class_ Base)

3.4.3.12. Class_Base_Block_Keys //The core index class for shared memory management in the framework(Inherit from framework base class Class_ Base)

3.4.3.13. Class_Base_Block_UniqueIndex //Unique index class for shared memory management in the framework(Inherit from framework base class Class_ Base)

3.4.3.14. Class_Base_BlockContent //Block Content Classes in the Memory Block Data Structure of the Shared Memory Management Framework(Inherit from framework base class Class_ Base)

3.4.3.15. Class_Base_BlockEndFlag //The block end flag class in the memory block data structure of the shared memory management framework(Inherit from framework base class Class_ Base)

3.4.3.16. Class_Base_BlockHead //Block header classes in the memory block data structure of shared memory management in the framework(Inherit from framework base class Class_ Base)

3.4.3.17. Class_Base_Bootstrap //Adaptive Bootstrap Encapsulation Class for Framework(Inherit from framework base class Class_ Base)

3.4.3.18. Class_Base_Database //Database operation class of framework(Inherit from framework base class Class_ Base)

3.4.3.19. Class_Base_Document //The basic document class of the framework(Inherit from framework base class Class_ Base)

3.4.3.20. Class_Base_Document_Elf64 //Framework's executable file format (ELF64) document class(Inherit from Framework Basic Document Class_ Base_ Document)

3.4.3.21. Class_Base_Elf //Framework's executable file format (ELF64) auxiliary function class(Inherit from framework base class Class_ Base)

3.4.3.22. Class_Base_Elf64 //Framework's executable file format (ELF64) content operation class(Inherit from framework base class Class_ Base)

3.4.3.23. Class_Base_Elf64_Dyn //Dyn structure class in the executable file format (ELF64) data structure of the framework(Inherit from framework base class Class_ Base)

3.4.3.24. Class_Base_Elf64_File_Header //ELF64 file header content operation class for framework(Inherit from framework base class Class_ Base)

3.4.3.25. Class_Base_Elf64_Program //Framework's executable file format (ELF64) Content program table operation class(Inherit from framework base class Class_ Base)

3.4.3.26. Class_Base_Elf64_Program_Header //Framework's executable file format (ELF64) Content program header table operation class(Inherit from framework base class Class_ Base)

3.4.3.27. Class_Base_Elf64_Rel //The Rel structure class in the executable file format (ELF64) data structure of the framework(Inherit from framework base class Class_ Base)

3.4.3.28. Class_Base_Elf64_Rela //The Rela structure class in the executable file format (ELF64) data structure of the framework(Inherit from framework base class Class_ Base)

3.4.3.29. Class_Base_Elf64_Section //Framework's executable file format (ELF64) Content section table operation class(Inherit from framework base class Class_ Base)

3.4.3.30. Class_Base_Elf64_Section_Header //Framework's executable file format (ELF64) Content section header table operation class(Inherit from framework base class Class_ Base)

3.4.3.31. Class_Base_Elf64_Section_Shstrtab //Framework's executable file format (ELF64) string section operation class(Inherit from framework base class Class_ Base)

3.4.3.32. Class_Base_Elf64_Sym //The Sym structure class in the executable file format (ELF64) data structure of the framework(Inherit from framework base class Class_ Base)

3.4.3.33. Class_Base_Error //Custom error base class for framework(Inherit from framework base class Class_ Base)

3.4.3.34. Class_Base_Exception //Custom exception base class for framework(Inherit from framework base class Class_ Base)

3.4.3.35. Class_Base_Extension //Framework Extension Processing Function Class(Inherit from framework base class Class_ Base)

3.4.3.36. Class_Base_File //File processing function class of the framework(Inherit from framework base class Class_ Base)

3.4.3.37. Class_Base_Format //The content format processing function class of the framework(Inherit from framework base class Class_ Base)

3.4.3.38. Class_Base_Lock //Framework's Concurrent Lock Processing Function Class(Inherit from framework base class Class_ Base)

3.4.3.39. Class_Base_Log //The log processing function class of the framework(Inherit from framework base class Class_ Base)

3.4.3.40. Class_Base_Memory //Shared Memory Processing Function Class of the Framework(Inherit from framework base class Class_ Base)

3.4.3.41. Class_Base_RawSocket //The original socket processing function class of the framework(Inherit from framework base class Class_ Base)

3.4.3.42. Class_Base_Report //Report processing function class of framework(Inherit from framework base class Class_ Base)

3.4.3.43. Class_Base_Request //Framework's network request data processing function class(Inherit from framework base class Class_ Base)

3.4.3.44. Class_Base_Response //Framework's network response data processing function class(Inherit from framework base class Class_ Base)

3.4.3.45. Class_Base_Security //Security processing function class of the framework(Direction of encryption and decryption)(Inherit from framework base class Class_ Base)

3.4.3.46. Class_Base_Security_Rsa //Security processing function class of the framework(Asymmetric encryption and decryption direction)(Inherit from framework base class Class_ Base)

3.4.3.47. Class_Base_Shell //Shell environment processing function class of the framework(Inherit from framework base class Class_ Base)

3.4.3.48. Class_Base_Socket //Network socket processing function class of the framework(Inherit from framework base class Class_ Base)

3.4.3.49. Class_Controller_Build //Project Construction Controller Class for Framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.50. Class_Controller_Chat //The instant messaging function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.51. Class_Controller_Clear //The cleaning function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.52. Class_Controller_Cookie //COOKIE Session Function Controller Class for Framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.53. Class_Controller_Database //Database processing function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.54. Class_Controller_Default //The default functional controller class for the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.55. Class_Controller_Elf //ELF format content processing function controller class for the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.56. Class_Controller_File //The file processing function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.57. Class_Controller_File_Download //Framework's file download function controller class(Inherited from the framework's controller base class Class_ Controller)

3.4.3.58. Class_Controller_File_Editor //The file editing function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.59. Class_Controller_File_Explorer //The file browsing function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.60. Class_Controller_File_Search //The file search function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.61. Class_Controller_File_Upload //The file upload function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.62. Class_Controller_Guide //Reference Materials for Framework Function Controller Class(Inherited from the framework's controller base class Class_ Controller)

3.4.3.63. Class_Controller_Index //The homepage function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.64. Class_Controller_Init //The initialization function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.65. Class_Controller_Login //Framework login function controller class(Inherited from the framework's controller base class Class_ Controller)

3.4.3.66. Class_Controller_Logout //Framework's logout function controller class(Inherited from the framework's controller base class Class_ Controller)

3.4.3.67. Class_Controller_Map //Map function controller class for framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.68. Class_Controller_Memory //The shared memory management function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.69. Class_Controller_PenetrationTestCommands //Penetration testing reference materials for frameworks, functional controller classes(Inherited from the framework's controller base class Class_ Controller)

3.4.3.70. Class_Controller_ProxyShell //Proxy Shell Environment Function Controller Class for Framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.71. Class_Controller_Report //The reporting function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.72. Class_Controller_Scan //Scan function controller class for framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.73. Class_Controller_Security //The security function controller class of the framework(Direction of encryption and decryption)(Inherited from the framework's controller base class Class_ Controller)

3.4.3.74. Class_Controller_Server //Framework's Service Environment Function Controller Class(Inherited from the framework's controller base class Class_ Controller)

3.4.3.75. Class_Controller_Session //The server session function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.76. Class_Controller_Shell //Shell Environment Function Controller Class for Framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.77. Class_Controller_Test //Penetration testing reference materials for frameworks, functional controller classes(Inherited from the framework's controller base class Class_ Controller)

3.4.3.78. Class_Controller_User //The account management function controller class of the framework(Inherited from the framework's controller base class Class_ Controller)

3.4.3.79. Class_Controller_Wget //Online download function of framework controller class(Inherited from the framework's controller base class Class_ Controller)

3.4.3.80. Class_Operate_Build //Building module classes in the formal environment of the framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.81. Class_Operate_ChatMemory //Framework's instant messaging function shared memory module class(Inherited from the framework's operate base class Class_Operate)

3.4.3.82. Class_Operate_ChatServer //The instant messaging server module class of the framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.83. Class_Operate_File //File processing module class for framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.84. Class_Operate_ProxyShell //Proxy Shell Environment Module Class for Framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.85. Class_Operate_Scan //The scanning function module class of the framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.86. Class_Operate_SocketServerShell //Server Shell Environment Module Class for Framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.87. Class_Operate_SocketShell //The server side rebound shell environment module class of the framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.88. Class_Operate_User //The account management module class of the framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.89. Class_Operate_Web //Web Environment Module Class for Framework(Inherited from the framework's operate base class Class_Operate)

3.4.3.90. Class_View_Default //Default Module View Class for Framework(Inherit from the framework's view base class Class_ View)

3.4.3.91. Class_View_Init //The Init module view class of the framework(Inherit from the framework's view base class Class_ View)

3.4.3.92. Class_View_Top //General header content view class for frameworks(Inherit from the framework's view base class Class_ View)

3.4.3.93. Class_View_Bottom //General bottom content view class for frameworks(Inherit from the framework's view base class Class_ View)

3.4.3.94. Class_View_Init_User_Info //Framework's' User_Info 'module menu view class(Inherit from the framework's view base class Class_ View)

3.4.3.95. Class_View_Login //Framework's Login Module View Class(Inherit from the framework's view base class Class_ View)

3.4.3.96. Class_View_Memory //The Memory module view class of the framework(Inherit from the framework's view base class Class_ View)

3.4.3.97. Class_View_Build_Menu //Build Module Menu View Class for Framework(Inherit from the framework's view base class Class_ View)

3.4.3.98. Class_View_Chat_Menu //The Chat module menu view class of the framework(Inherit from the framework's view base class Class_ View)

3.4.3.99. Class_View_Cookie_Menu //Cookie module menu view class for framework(Inherit from the framework's view base class Class_ View)

3.4.3.100. Class_View_Database_Menu //Database module menu view class for framework(Inherit from the framework's view base class Class_ View)

3.4.3.101. Class_View_Elf_Menu //Elf module menu view class for framework(Inherit from the framework's view base class Class_ View)

3.4.3.102. Class_View_File_Menu //The File module menu view class of the framework(Inherit from the framework's view base class Class_ View)

3.4.3.103. Class_View_Guide_CommonCommand_Menu //Framework's Guide_ CommonCommand module menu view class(Inherit from the framework's view base class Class_ View)

3.4.3.104. Class_View_Guide_Menu //Framework's Guide Module Menu View Class(Inherit from the framework's view base class Class_ View)

3.4.3.105. Class_View_Guide_PenetrationTestCommands_Menu //Framework's Guide_ PenetrationTestCommands module menu view class(Inherit from the framework's view base class Class_ View)

3.4.3.106. Class_View_Memory_Menu //Framework's Memory Module Menu View Class(Inherit from the framework's view base class Class_ View)

3.4.3.107. Class_View_ProxyShell_Menu //The ProxyShell module menu view class for the framework(Inherit from the framework's view base class Class_ View)

3.4.3.108. Class_View_Report_Menu //Framework's Report Module Menu View Class(Inherit from the framework's view base class Class_ View)

3.4.3.109. Class_View_Scan_Menu //Scan module menu view class for framework(Inherit from the framework's view base class Class_ View)

3.4.3.110. Class_View_Security_Menu //Framework's Security Module Menu View Class(Inherit from the framework's view base class Class_ View)

3.4.3.111. Class_View_Server_Menu //Server Module Menu View Class for Framework(Inherit from the framework's view base class Class_ View)

3.4.3.112. Class_View_Session_Menu //Session module menu view class of the framework(Inherit from the framework's view base class Class_ View)

3.4.3.113. Class_View_Shell_Menu //Shell Module Menu View Class for Framework(Inherit from the framework's view base class Class_ View)

3.4.3.114. Class_View_User_Menu //Framework's User Module Menu View Class(Inherit from the framework's view base class Class_ View)

3.4.3.115. Class_View_Wget_Menu //Framework's Wget module menu view class(Inherit from the framework's view base class Class_ View)

3.4.4. introduction to featured functions

Phpsploit-Framework software pioneered a shell communication environment based on raw sockets (proxyshell), which can bypass more than 80% of the firewall rules when conducting CTF competitions or penetration tests. Because the shell communication environment communicates directly based on IPv6 protocol, the rules of packet filtering based on transport layer can be invalidated. Based on the extended header feature of IPv6 protocol, the shell environment using IPv6 protocol for communication can realize the proxy forwarding function for shell command requests, and can break through the network defense environment based on IP address restrictions. Comprehensive evaluation shows that the shell communication environment based on the network layer will have a higher level of concealment when conducting CTF competition or penetration test. If properly used, this shell communication environment based on raw sockets can become an effective weapon for CTF competition or penetration test behavior!

A set of instant messaging system (Chat) based on c/s architecture has been integrated into Phpsploit-Framework software. Any environment where Phpsploit-Framework software is deployed can run the instant messaging system of Phpsploit-Framework software (note that the service side operation of this instant messaging system depends on shmop extension). The client of this instant messaging system is written in Java language, and the server is written in PHP language. The communication parties use TCP protocol for communication, The communication data has been encrypted in a relatively secure RSA asymmetric way (however, it should be noted that before you formally create and use the project program files of the production environment, you should modify the communication key information in the source code files of the client and server. Warning: it is not safe to use the original RSA key for data encryption! This may cause the communication data to be hijacked and decrypted successfully! The RSA key information of the client is located in the source code directory of the instant messaging project "Chat.system.lib.security.rsa" class. The RSA key information of the server is located in the "class_base_security_rsa" class of Phpsploit-Framework software). When the public chat environment may be attacked by social engineering, using the instant messaging function of Phpsploit-Framework software can also become a private choice for team cooperation and attack and defense cooperation.

Phpsploit-Framework software integrates a variety of shell environments internally. Compared with the traditional penetration test shell environment, most of the built-in shells of Phpsploit-Framework software integrate the function of secure communication. Webshell has used dynamic key technology to encrypt the communication data in real time, and the data key used in each communication is different. The servershell based on c/s architecture has also used dynamic token and dynamic key technology to encrypt the communication data. Proxyshell technology also uses token token based on rules and dynamic encryption technology to improve the communication security of shell environment. At the same time, proxyshell also adopts white list mechanism to minimize the risk of malicious attack on proxyshell.

The database management function (database) built in the Phpsploit-Framework software has dynamically encrypted the transmitted data (the encryption key used for each communication is different). It can reduce the risk of communication data hijacking and malicious attack to a greater extent. This database management function module is subdivided into two sub modules: query and exec. According to the different characteristics of the module, the execution results of SQL statements in the database management function module will be reflected in different forms (for example, for multiple queries or update statements, the execution results corresponding to the statements will be displayed in the form of sets).

The self-contained elf format file parsing function of Phpsploit-Framework software provides relevant support for content parsing of ELF format binary content. Phpsploit-Framework software users can use this function for security analysis of binary content. This function can automatically extract the file header, program header table, section header table, program table and section table data of ELF format files, and display them in the specified format. When conducting CTF competition and penetration test, this function can provide certain data analysis support for overflow fuzzy test. Combined with the shared memory management function provided by Phpsploit-Framework software, you can further verify whether overflow class or remote code execution class vulnerabilities exist. When conducting penetration testing, it is very necessary to discover and effectively repair the security vulnerabilities in their own systems before malicious hackers! Comprehensive evaluation, the elf format file parsing function of phpsploit framework software is a very practical hard core capability with a wide range of application scenarios.

The internal shared memory management function (memory) of Phpsploit-Framework software provides the ability to manage the shared memory resources in the operating system. The number of shared memory resources that can be found and accessed by this shared memory management function is closely related to the permissions when running Phpsploit-Framework software. This function provides the ability to view, create, update, and clear shared memory resources. It can be used alone or in combination with the elf format file parsing function of Phpsploit-Framework software for fuzzy testing of binary security classes. This shared memory management function also manages some shared memory resources bound by the Phpsploit-Framework software itself. It can be said that this function is one of the very important core functions in Phpsploit-Framework software.

Compared with the file download function in the traditional penetration testing environment, the built-in file online download (WGet) function of Phpsploit-Framework software has more advantages. First of all, the online file download function of Phpsploit-Framework software can provide real-time display of download progress, which is conducive to Phpsploit-Framework software users' real-time observation of file download progress. At the same time, after the files downloaded using the file online download function are successfully saved on the server, a file name with significant identification will be generated (with the word.Phpsploit in the file name), which can facilitate the corresponding cleaning after the completion of penetration test and security audit.

The file management function (file) built in the Phpsploit-Framework software realizes the simple file management feature. The search function provided in the file management function can easily search the specified directory or file in the specified path; The browsing function provided in the file management function can easily view the contents of the directory under the specified path; The reading function provided in the file management function allows you to see the truth inside the file (for the display of file content, there are two modes that can be switched, namely binary format and plain text format. For large volume files, paging browsing is supported, which can basically meet the analysis requirements for slightly larger files); The file creation function provided in the file management function can help you create the specified file under the specified path. At the same time, in order to facilitate management (for example, after the penetration test and security audit are completed, the corresponding file cleaning is carried out), the file name of the new file created by Phpsploit-Framework software will contain the words ".phpsploit". The file editing function provided in the file management function allows you to easily edit the corresponding files created or uploaded through the phpsploit framework software (but please note that editing large files may lead to server load increase, web page blocking, browser unresponsiveness, etc., please be careful to avoid editing large files); The file upload function provided in the file management function allows you to easily upload the corresponding files to the server space (but please be careful! Please do not upload files that you cannot effectively control to the server space, which may cause great risks to the server space). When the file is uploaded successfully, The file name of the newly uploaded file will contain the words ".phpsploit" (this setting is mainly for you to easily clean up the uploaded file after the completion of penetration test and security audit); The file deletion function provided in the file management function can quickly delete the relevant files downloaded, uploaded and created directly by Phpsploit-Framework software users using Phpsploit-Framework software; The file cleanup function provided in the file management function enables one click cleanup of related files (for example, related files with the words ".phpsploit"). Note that for security reasons, Phpsploit-Framework software is designed to only edit and delete the relevant files that Phpsploit-Framework software users directly download, upload and create using Phpsploit-Framework software! For relevant files that are not directly downloaded, uploaded or created by Phpsploit-Framework software, Phpsploit-Framework software will not be able to successfully edit, delete and other behaviors.

The built-in scanning function (scan) of Phpsploit-Framework software can help Phpsploit-Framework software users conveniently carry out web site survival status detection and host port open status detection. At the same time, a more distinctive design is that the built-in scanning function (scan) of the Phpsploit-Framework software also provides a very user-friendly experience and practical functions for security operation and maintenance personnel. By using the sample verification function provided in the scan function, the security operation and maintenance personnel only need to enter the corresponding paths of the sample directory and sampling directory respectively, and then click the "start scan samplerproof" button, You can perform content matching verification on the sampling directory based on the content of the specimen catalog The main matching contents include: 1. whether the number of subdirectories and files in the specimen directory is the same as the number of subdirectories and files in the sampling directory; 2. whether there are directories or files that do not exist in the specimen directory and its corresponding subdirectories in the sampling directory and its subordinate subdirectories; 3. whether the directories or files that exist in the specimen directory and its subordinate subdirectories are not found in the sampling directory and its corresponding subdirectories; 4. Does the file size in the specimen directory and its subdirectories differ from that in the sampling directory and its subdirectories, and does the file have different md5 or sha1 checksums. At the same time, the sample verification function (tapperproof) will provide a warning prompt (usually related information in red) for any abnormal content found!

The built-in encryption and decryption function (security) of the Phpsploit-Framework software provides a large number of practical online encryption and decryption tools. Phpsploit-Framework software users no longer need to search for corresponding data encryption and decryption tools through various channels! By using the built-in encryption and decryption function (security) of the Phpsploit-Framework software, you can complete encryption and decryption operations for the vast majority of general data (commonly used encryption and decryption operations include URL encode/code, base64 encode/code, crypt encode/code, openssl encode/code, md5, sha1, hash, etc.)!

The built-in report function of the Phpsploit-Framework software can very conveniently assist users of the Phpsploit-Framework software in creating vulnerability report documents online (in xls format). Through this report function, Phpsploit-Framework software users can easily create vulnerability reports without installing Excel software! When the vulnerability report is completed, users of the Phpsploit-Framework software can download the report file automatically generated based on the content of the vulnerability report online through the export vulnerability report function that comes with the report function.

The guidance manual function provided internally by the Phpsploit-Framework software integrates a large amount of penetration testing/security operation and maintenance documentation that the author of the Phpsploit-Framework software has been collecting and organizing for over a year (including the translation of a large amount of English materials, which consumes a lot of energy and time of the Phpsploit-Framework software author). In the guide function, specific usage of various commands is integrated. This includes utility related command usage in fields such as information collection, vulnerability analysis, web programs, database evaluation, password security, wireless security, reverse engineering, vulnerability exploitation, sniffing/spoofing, permission maintenance, digital forensics, security reporting, social engineering, etc.

3.4.5. Matters needing attention

3.4.5.1. Before officially using the Phpsploit Framework software, users of the Phpsploit Framework software should carefully read the entire content of the user agreement and disclaimer, and follow the content stipulated in the user agreement to use the Phpsploit Framework software in a legal and compliant manner! Any unauthorized penetration testing and security auditing activities using the Phpsploit Framework software are explicitly prohibited by the author of the Phpsploit Framework software! If a user of the Phpsploit Framework software violates the user agreement and uses the Phpsploit Framework software for any illegal purpose, the author of the Phpsploit Framework software shall not be held responsible! At the same time, the author of the Phpsploit Framework software will hold all legal responsibilities of Phpsploit Framework software users for violating the user agreement in accordance with the law!

3.4.5.2. Users of phpploit-framework software should be aware of the difference between privileged accounts and regular accounts when using phpploit-framework software (even if they have the same account name)! Privileged accounts are usually used in specific situations (for example, during the initial installation of the Phpsploit-Framework software, for the creation of normal accounts, and for the assignment of dynamic passwords and md5_token tokens). Ordinary account, mainly used for php framework software login behavior and function use behavior! Therefore, as a user of the Phpsploit-Framework software, you must remember the public account password and md5_token information generated during each initial installation of the Phpsploit-Framework software. Otherwise, after the initial installation of the phpploit-framework software is complete, you may fail to log in to the phpploit-framework software because you forget the public account password and md5_token information. If this happens, you can only use a privileged account to call the /clear route of the Phpsploit-Framework software from the command line environment of the system where the Phpsploit-Framework software resides to resolve the problem! If it still fails, you can only try to execute ipcrm -M 0x5d8a0000 and ipcrm -M 0x5d8a0001 to fix the problem! But manipulating shared memory resources directly in the environment where phpploit-framework software is located is risky! The shared memory resources corresponding to KEY 0x5d8a0000 and KEY 0x5d8a0001 are used to store public account information of the phploit-framework software. However, it is also possible for other software to occupy these two shared memory resources (Although the chances of this happening are not very high! In addition, unless the software that occupies the shared memory resources KEY

0x5d8a0000 and KEY 0x5d8a0001 has an access permission setting error for these two shared memory resources. Otherwise, phpploit-framework cannot read or write shared memory resources KEY 0x5d8a0000 and KEY 0x5d8a0001, and the phpploit-framework fails to be initialized. KEY 0x5d8a0000 and KEY 0x5d8a0001 used by the phpploit-framework software have corresponding feature information. You can perform operations as required. In normal cases, the phploit-framework uses KEY 0x5d8a0000 to share memory size 32 bytes and memory access permission 660(octal). The KEY 0x5d8a0001 used by phpploit-framework has a shared memory size of 1048712 bytes and memory access of 660(octal). If you are not sure whether shared memory resources are cleared correctly, contact the system administrator of the phploit-framework environment. In general, the worst outcome is that you may need to work with the system administrator of the environment where the phpploit-framework software is located to resolve the issue by restarting the server in the environment where the phpploit-framework software is located! But this is the worst! In normal cases, clear shared memory resources KEY 0x5d8a0000 and KEY 0x5d8a0001 used by the phploit-framework software to resolve the problem. In a Web environment, the phpploit-framework software uses Session information to verify the communication between the browser and the phpploit-framework software on the Web server. In the Cli environment, the Phpsploit-Framework software uses the md5_token created during the initial installation of the Phpsploit-Framework software for communication authentication. Authentication information of common accounts is saved in the shared memory. Therefore, if the phpplot-framework software is running in an environment where shmop extensions are not enabled, the phpplot-framework software will use privileged accounts to be compatible with normal accounts. In this case, the account name and password of the privileged account and the common account are the same.

3.4.5.3. Users of the Phpsploit-Framework software, when trying to build a production application project file using the /build function of the Phpsploit-Framework software, please note that, Phpsploit-Framework software users need to set the values of the constant PRIVILEGE_USER_MODULE_USER and PRIVILEGE_USER_MODULE_USER to the names and passwords of the privileged accounts in the production environment, respectively. The Phpsploit-Framework users can obtain the correct production environment privileged account name and password by accessing the /user/create_production_privilege_user_password route. This route is set exclusively to create the correct production environment privileged account name and corresponding password information.

3.4.5.4. Phpsploit-Framework software users before formally creating and using production environment project program files, The communication key information located in the client (java) source file in the instant messaging system and the communication key information located in the Phpsploit-Framework software (php) source file should be synchronously modified! Note that data encrypted using the client's RSA public key needs to be decrypted using the client's RSA private key! The data encrypted using the RSA public key on the server needs to be decrypted using the RSA private key on the server. Tip, the client and the server are stored in the corresponding source code of the client and the server RSA public and private key information, you need to synchronize them to update, in order to keep the RSA key information consistent! Warning: Using the original RSA key for data encryption is not secure! This may result in the communication data being successfully decrypted after being hijacked! Client RSA key information, located in the instant communication System of the project source directory "Chat. System. Lib. Security. RSA" class. The RSA key information on the server is stored in the Class_Base_Security_Rsa class of the Phpsploit-Framework software.

3.4.5.5. Phpsploit-Framework Software Users should choose a secure network environment when using Phpsploit-Framework software. Among the application layer protocols based on Web applications, HTTPS is far more secure than HTTP. Located in the middle of TCP and HTTP protocol SSL layer, you can maximize the security of communication data! However, you still need to watch out for man-in-the-middle spoofing attacks from fake gateways!

3.4.5.6. Some of the content of this technical white paper document is automatically generated by translation software, not 100% written by manual means, so there may be some content errors! If you find relevant errors in the content of this file, please contact the Phpsploit-Framework software author to check and correct the relevant content, thank you!