Phpsploit-Framework 软件技术白皮书

一、软件简介

phpsploit-framework 是一个开源的 CTF 框架和漏洞利用开发库。它使用 PHP 语言编写而成，专为渗透测试和安全审计行为而设计，旨在能够帮助道德黑客（渗透测试工程师、IT 审计师、安全研发工程师等）更加高效快捷地开展漏洞挖掘与安全审计工作。

软件项目地址是 https://github.com/huc0day/phpsploit-framework ， 使用 github 作为开源代码的存储仓库。

二、软件作者

phpsploit-framework 软件作者（网名"huc0day"），是一名致力于开源事业的道德黑客。历任多家互联网公司的研发工程师、软件架构师、技术总监、信息安全官、高级 IT 审计师等职务，擅长渗透测试、逆向工程、Web 安全、二进制安全、安全类软件研发等领域的相关工作。曾为早期的红客大联盟成员，从 2004 年起至今，分别设计有网络安全编程框架、内存型数据库核心引擎、GDSC 数据安全通信协议、服务器入侵报警系统等原创作品。 联系邮箱：huc0day@outlook.com 。

三、软件使用

1、phpsploit-framework 软件的运行环境（php5.6+的 Web 或者 Cli 环境）

（1） PHP 语言版本（5.6+）

（2） PHP 扩展：bcmath、openssl、pcntl、pdo_mysql、shmop、curl 等

注意：当部分扩展未安装时，phpsploit-framework 软件的部分功能将不可用。

（3） 操作系统环境：Linux 系列

（4） Apache2 或 Nginx 或 Cli 环境

注意：phpsploit-framework 软件的部分功能仅能运行于 root 权限与 cli 模式。

（5） Mysql 或 Mariadb

（6） 除 IE 浏览器之外的大部分浏览器（如：谷歌浏览器、火狐浏览器等）

2、phpsploit-framework 软件的即时通信客户端的运行环境（openjdk-21）

（1） JAVA 语言版本（openjdk-21）

（2） JAVA 第三方包：fastjson2-2.0.6.jar、jedis-4.2.3.jar、javacv-1.4.1.jar

（3） 操作系统环境：Linux 系列

3、phpsploit-framework 软件的三种运行模式（开发模式、测试模式、生产模式）：

（1）开发模式

在此模式下，包含完整的框架文件，你可以方便地在原有框架内容基础上进行扩展、优化和改进，以使其更加地适应于您当前的场景需求！

（2）测试模式

在此模式下，同样包含完整的框架文件，不同之处在于将使用正式模式下的相关配置进行软件运行。在此模式下，你可以较为方便地对软件运行情况进行调试，并对软件运行过程中发现的 BUG 问题进行有效修复！

（3）生产模式

在此模式下，框架整体被打包成单一且无其它依赖文件的项目程序文件（文件名称根据项目程序文件的版本类型（精简版本（Lite）和完整版本（Full））不同而

存在一定差异，项目程序文件名格式为 build_<lite/full>_<18 位长整型数字>.phpsploit.php，此模式下的项目程序文件被广泛用于得到合法授权前提下的渗透测试与安全审计等行为。

4、phpsploit-framework 软件的相关功能介绍

（1）关键功能路由介绍

/home //家目录
/user //账户目录
/user_info //显示当前登陆的账户信息
/create_production_account //创建特权账户信息
/unsubscribe //注销当前登陆的账户信息
/logout //退出当前登陆
/server //服务器环境目录
/server_info //显示服务器环境信息
/session //服务端会话环境目录
/session_info //显示服务端会话信息
/cookie //客户端会话环境目录
/cookie_info //显示客户端会话信息
/build //构建项目文件
/encode_build //构建加密的文件
/decode_build //构建解密的文件
/debug //切换调试模式
/guide //参考资料目录
/guide/common_commands //常用命令介绍模块
/guide/user_commands //账户相关命令介绍模块
/guide/elf_commands //ELF 格式内容操作相关命令介绍模块
/guide/system_commands //操作系统相关命令介绍模块
/guide/file_commands //文件操作相关命令介绍模块
/guide/hardware_commands //硬件操作相关命令介绍模块
/guide/software_commands //软件操作相关命令介绍模块
/guide/network_commands //网络操作相关命令介绍模块
/guide/firewall_commands //防火墙操作相关命令介绍模块
/guide/webserver_commands //Web 服务器操作相关命令介绍模块
/guide/docker_commands //Docker 容器操作相关命令介绍模块
/guide/penetration_test_commands //渗透测试相关命令介绍模块
/guide/penetration_test_commands/information_gathering /渗透测试之信息收集相关命令介绍模块
/guide/penetration_test_commands/vulnerability_analysis //渗透测试之漏洞分析相关命令介绍模块
/guide/penetration_test_commands/web_program //渗透测试之 Web 程序相关命令介绍模块
/guide/penetration_test_commands/database_evaluation //渗透测试之数据库评估相关命令介绍模块
/guide/penetration_test_commands/password_attack //渗透测试之密码攻击相关命令介绍模块

/guide/penetration_test_commands/wireless_attacks //渗透测试之无线攻击相关命令介绍模块

/guide/penetration_test_commands/reverse_engineering //渗透测试之逆向工程相关命令介绍模块

/guide/penetration_test_commands/vulnerability_exploitation //渗透测试之漏洞利用相关命令介绍模块

/guide/penetration_test_commands/sniff_deception //渗透测试之嗅探/欺骗相关命令介绍模块

/guide/penetration_test_commands/permission_maintenance //渗透测试之权限维持相关命令介绍模块

/guide/penetration_test_commands/data_forensics //渗透测试之数字取证相关命令介绍模块

/guide/penetration_test_commands/reporting //渗透测试之报告工具相关命令介绍模块

/guide/penetration_test_commands/social_engineering //渗透测试之社会工程相关命令介绍模块

/security //加解密功能模块

/security/url //加解密功能之 url 加解密模块

/security/base64 //加解密功能之 base64 加解密模块

/security/sha1 //加解密功能之 sha1 加解密模块

/security/md5 //加解密功能之 md5 加解密模块

/security/crc32 //加解密功能之 crc32 加解密模块

/security/crypt //加解密功能之 crypt 加解密模块

/security/openssl //加解密功能之 openssl 加解密模块

/security/hash //加解密功能之 hash 加解密模块

/security/password_hash //加解密功能之 password_hash 加解密模块

/security/sodium //加解密功能之 sodium 加解密模块

/security/hash_hmac //加解密功能之 hash_hmac 加解密模块

/memory //共享内存管理模块

/memory/system //共享内存管理之系统管理模块

/memory/search //共享内存管理之共享内存资源搜索模块

/memory/list //共享内存管理之共享内存资源列表模块

/memory/add //共享内存管理之共享内存资源添加模块

/memory/clear //共享内存管理之共享内存资源清理模块

/database //数据库管理模块

/database/query //数据库管理之数据查询模块

/database/exec //数据库管理之数据更新模块

/file //文件管理模块

/file/search //文件管理之文件搜索模块

/file/explorer //文件管理之文件浏览模块

/file/create //文件管理之文件创建模块

/file/upload //文件管理之文件上传模块

/file/clear //文件管理之文件清理模块

/scan //扫描管理模块

/scan/webs //扫描管理之 Web 站点存活情况检测模块
/scan/domain //扫描管理之主机端口开放情况检测模块
/scan/tamperproof //扫描管理之异常样本检测与预警模块
/wget //文件在线下载模块
/elf //ELF 格式内容解析模块
/elf/elf64 //ELF 格式内容解析之 ELF64 格式文件解析模块
/elf/elf_h //ELF 格式内容解析之 C 语言源码定义阅读模块
/shell //Shell 环境管理模块
/shell/web_shell //Shell 环境管理之 WebShell 环境模块
/shell/server_shell //Shell 环境管理之基于 C/S 架构的服务端模块
/shell/server_shell_client //Shell 环境管理之基于 C/S 架构的客户端模块
/shell/reverse_shell //Shell 环境管理之反弹 Shell 环境模块
/shell/background_shell //Shell 环境管理之反弹 Shell 环境模块
/shell/proxy_shell //Shell 环境管理之代理 Shell 环境模块
/shell/proxy_shell/create_session_id //代理 Shell 环境之创建认证信息
/shell/proxy_shell/clear_session_id //代理 Shell 环境之清理认证信息
/shell/proxy_shell/send //代理 Shell 环境之发送 Shell 命令
/shell/proxy_shell/receive //代理 Shell 环境之接收 Shell 命令执行结果
/shell/proxy_shell/listen //代理 Shell 环境之监听并执行 Shell 命令
/chat //即时通信管理模块
/chat/server_chat //即时通信管理之基于 C/S 架构的即时通信服务端模块
/chat/reverse_chat //即时通信管理之基于反弹连接技术的客户端模块
/report //报告管理模块
/report/create_vulnerability_report //报告管理之创建报告模块
/report/edit_vulnerability_report //报告管理之编辑报告模块
/report/show_vulnerability_report //报告管理之查看报告模块
/report/export_vulnerability_report //报告管理之导出报告模块
/report/clear_vulnerability_report //报告管理之清理报告模块
/clear //重置 Phpsploit-Framework 软件的登陆状态与相关环境信息
/logout //退出登陆

（2）关键功能接口介绍

Interface_Root //框架的基础接口（框架根接口）
Interface_Base //框架的基础功能接口库根接口（继承自 Interface_Root）
Interface_Operate //框架的模块功能接口库根接口（继承自 Interface_Root）
Interface_Controller //框架的控制器接口库根接口（继承自 Interface_Root）
Interface_View //框架的视图表示层接口库根接口（继承自 Interface_Root）
Interface_Main //框架的程序调用接口（继承自 Interface_Root）
Interface_Base_Block //框架的共享内存管理的基础接口（继承自框架接口 Interface_Base）
Interface_Base_Block_Data //框架的共享内存管理的数据接口（继承自框架接口 Interface_Base）
Interface_Base_Block_Indexes //框架的共享内存管理的索引列表接口（继承自框架接口 Interface_Base）

Interface_Base_Block_IndexesItem //框架的共享内存管理的索引列表子项接口（继承自框架接口 Interface_Base）

Interface_Base_Block_Keys //框架的共享内存管理的核心索引接口（继承自框架接口 Interface_Base）

Interface_Base_Block_UniqueIndex //框架的共享内存管理的唯一索引接口（继承自框架接口 Interface_Base）

Interface_Base_BlockContent //框架的共享内存管理的内存块数据结构中的块内容接口（继承自框架接口 Interface_Base）

Interface_Base_BlockContentType //框架的共享内存管理的内存块数据结构中的块内容类型接口（继承自框架接口 Interface_Base）

Interface_Base_BlockEndFlag //框架的共享内存管理的内存块数据结构中的块结束标志接口（继承自框架接口 Interface_Base）

Interface_Base_BlockHead //框架的共享内存管理的内存块数据结构中的块头部接口（继承自框架接口 Interface_Base）

Interface_Base_BlockHeadEndFlag //框架的共享内存管理的内存块数据结构中的块头部结束标志接口（继承自框架接口 Interface_Base）

Interface_Base_BlockKey //框架的共享内存管理的内存块数据结构中的块索引字段接口（继承自框架接口 Interface_Base）

Interface_Base_BlockMode //框架的共享内存管理的内存块数据结构中的块模式字段接口（继承自框架接口 Interface_Base）

Interface_Base_BlockName //框架的共享内存管理的内存块数据结构中的块名字字段接口（继承自框架接口 Interface_Base）

Interface_Base_BlockReserved //框架的共享内存管理的内存块数据结构中的块保留字段接口（继承自框架接口 Interface_Base）

Interface_Base_BlockSize //框架的共享内存管理的内存块数据结构中的块大小字段接口（继承自框架接口 Interface_Base）

Interface_Base_BlockStatus //框架的共享内存管理的内存块数据结构中的块状态字段接口（继承自框架接口 Interface_Base）

Interface_Base_BlockType //框架的共享内存管理的内存块数据结构中的块类型段接口（继承自框架接口 Interface_Base）

Interface_Base_Document //框架的文档处理功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Error //框架的错误处理功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Exception //框架的异常处理功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_File //框架的文件处理功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Format //框架的格式转换功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_FormatType //框架的格式类型定义基础接口（继承自框架接口 Interface_Base）

Interface_Base_Lock //框架的并发锁功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Memory //框架的共享内存功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_RawSocket //框架的网络原始套接字功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Request //框架的网络请求功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_ResourceType //框架的资源类型定义基础接口（继承自框架接口 Interface_Base）

Interface_Base_Response //框架的网络应答功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Security //框架的安全功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Security_Rsa //框架的非对称加密功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Shell //框架的 Shell 环境功能基础接口（继承自框架接口 Interface_Base）

Interface_Base_Socket //框架的网络套接字功能基础接口（继承自框架接口 Interface_Base）

Interface_Controller_ChatServer //框架的聊天服务器功能控制器基础接口（继承自框架接口 Interface_Controller）

Interface_Operate_ChatServer //框架的聊天服务器功能模块基础接口（继承自框架接口 Interface_Operate）

Interface_Operate_User //框架的用户功能模块基础接口（继承自框架接口 Interface_Operate）

（3）关键功能类介绍

Class_Root //框架的基础基类（继承自 PHP 语言自有基类：StdClass）

Class_Base //框架的基础功能类库根类（继承自框架根类：Class_Root）

Class_Operate //框架的模块功能类库根类（继承自框架根类：Class_Root）

Class_Controller //框架的控制器类库根类（继承自框架根类：Class_Root）

Class_VIew //框架的视图表示层类库根类（继承自框架根类：Class_Root）

Class_Main //框架的程序调用入口类（继承自框架根类 Class_Root）

Class_Base_Auth //框架的登陆验证类（继承自框架基础类 Class_Base）

Class_Base_Block //框架的共享内存管理的基础类（继承自框架基础类 Class_Base）

Class_Base_Block_Data //框架的共享内存管理的数据类（继承自框架基础类 Class_Base）

Class_Base_Block_Indexes //框架的共享内存管理的索引列表类（继承自框架基础类 Class_Base）

Class_Base_Block_IndexesItem //框架的共享内存管理的索引列表子项类（继承自框架基础类 Class_Base）

Class_Base_Block_Keys //框架的共享内存管理的核心索引类（继承自框架基础类 Class_Base）

Class_Base_Block_UniqueIndex //框架的共享内存管理的唯一索引类（继承自框架基础类 Class_Base）

Class_Base_BlockContent //框架的共享内存管理的内存块数据结构中的块内容类（继承自框架基础类 Class_Base）

Class_Base_BlockEndFlag //框架的共享内存管理的内存块数据结构中的块结束标志类（继承自框架基础类 Class_Base）

Class_Base_BlockHead //框架的共享内存管理的内存块数据结构中的块头部类（继承自框架基础类 Class_Base）

Class_Base_Bootstrap //框架的自适应 Bootstrap 封装类（继承自框架基础类 Class_Base）

Class_Base_Database //框架的数据库操作类（继承自框架基础类 Class_Base）

Class_Base_Document //框架的基础文档类（继承自框架基础类 Class_Base）

Class_Base_Document_Elf64 //框架的可执行文件格式（ELF64）文档类（继承自框架基础文档类 Class_Base_Document）

Class_Base_Elf //框架的可执行文件格式（ELF64）辅助功能类（继承自框架基础类 Class_Base）

Class_Base_Elf64 //框架的可执行文件格式（ELF64）内容操作类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Dyn //框架的可执行文件格式（ELF64）数据结构中的 Dyn 结构类（继承自框架基础类 Class_Base）

Class_Base_Elf64_File_Header //框架的 ELF64 文件头内容操作类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Program //框架的可执行文件格式（ELF64）内容程序表操作类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Program_Header //框架的可执行文件格式（ELF64）内容程序头表操作类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Rel //框架的可执行文件格式（ELF64）数据结构中的 Rel 结构类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Rela //框架的可执行文件格式（ELF64）数据结构中的 Rela 结构类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Section //框架的可执行文件格式（ELF64）内容节表操作类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Section_Header //框架的可执行文件格式（ELF64）内容节头表操作类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Section_Shstrtab //框架的可执行文件格式（ELF64）字符串节操作类（继承自框架基础类 Class_Base）

Class_Base_Elf64_Sym //框架的可执行文件格式（ELF64）数据结构中的 Sym 结构类（继承自框架基础类 Class_Base）

Class_Base_Error //框架的自定义错误基类（继承自框架基础类 Class_Base）

Class_Base_Exception //框架的自定义异常基类（继承自框架基础类 Class_Base）

Class_Base_Extension //框架的扩展处理功能类（继承自框架基础类

Class_Base）

Class_Base_File //框架的文件处理功能类（继承自框架基础类 Class_Base）

Class_Base_Format //框架的内容格式处理功能类（继承自框架基础类 Class_Base）

Class_Base_Lock //框架的并发锁处理功能类（继承自框架基础类 Class_Base）

Class_Base_Log //框架的日志处理功能类（继承自框架基础类 Class_Base）

Class_Base_Memory //框架的共享内存处理功能类（继承自框架基础类 Class_Base）

Class_Base_RawSocket //框架的原始套接字处理功能类（继承自框架基础类 Class_Base）

Class_Base_Report //框架的报表处理功能类（继承自框架基础类 Class_Base）

Class_Base_Request //框架的网络请求数据处理功能类（继承自框架基础类 Class_Base）

Class_Base_Response //框架的网络应答数据处理功能类（继承自框架基础类 Class_Base）

Class_Base_Security //框架的安全处理功能类（继承自框架基础类 Class_Base）

Class_Base_Security_Rsa //框架的安全处理功能类（非对称加密方向）

Class_Base_Shell //框架的 Shell 环境处理功能类（继承自框架基础类 Class_Base）

Class_Base_Socket //框架的网络套接字处理功能类（继承自框架基础类 Class_Base）

Class_Controller_Build //框架的项目构建控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Chat //框架的即时通信功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Clear //框架的清理功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Cookie //框架的 COOKIE 会话功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Database //框架的数据库处理功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Default //框架的默认功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Elf //框架的 ELF 格式内容处理功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_File //框架的文件处理功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_File_Download //框架的文件下载功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_File_Editor //框架的文件编辑功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_File_Explorer //框架的文件浏览功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_File_Search //框架的文件搜索功能控制器类（继承自框架的

控制器基类 Class_Controller）

Class_Controller_File_Upload //框架的文件上传功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Guide //框架的参考资料功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Index //框架的主页功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Init //框架的初始化功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Login //框架的登入功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Logout //框架的登出功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Map //框架的地图功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Memory //框架的共享内存管理功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_PenetrationTestCommands //框架的渗透测试参考资料功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_ProxyShell //框架的代理 Shell 环境功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Report //框架的报告功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Scan //框架的扫描功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Security //框架的安全功能控制器类（加解密方向）（继承自框架的控制器基类 Class_Controller）

Class_Controller_Server //框架的服务环境功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Session //框架的服务端会话功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Shell //框架的 Shell 环境功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Test //框架的渗透测试参考资料功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_User //框架的账户管理功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Controller_Wget //框架的在线下载功能控制器类（继承自框架的控制器基类 Class_Controller）

Class_Operate_Build //框架的正式环境构建模块类（继承自框架的模块基类 Class_Operate）

Class_Operate_ChatMemory //框架的即时通信功能共享内存模块类（继承自框架的模块基类 Class_Operate）

Class_Operate_ChatServer //框架的即时通信服务端模块类（继承自框架的模

块基类 Class_Operate）

Class_Operate_File //框架的文件处理模块类（继承自框架的模块基类 Class_Operate）

Class_Operate_ProxyShell //框架的代理 Shell 环境模块类（继承自框架的模块基类 Class_Operate）

Class_Operate_Scan //框架的扫描功能模块类（继承自框架的模块基类 Class_Operate）

Class_Operate_SocketServerShell //框架的服务端 Shell 环境模块类（继承自框架的模块基类 Class_Operate）

Class_Operate_SocketShell //框架的服务端反弹 Shell 环境模块类（继承自框架的模块基类 Class_Operate）

Class_Operate_User //框架的账户管理模块类（继承自框架的模块基类 Class_Operate）

Class_Operate_Web //框架的 Web 环境模块类（继承自框架的模块基类 Class_Operate）

Class_View_Default //框架的 Default 模块视图类（继承自框架的视图基类 Class_View）

Class_View_Init //框架的 Init 模块视图类（继承自框架的视图基类 Class_View）

Class_View_Top //框架的通用头部内容视图类（继承自框架的视图基类 Class_View）

Class_View_Bottom //框架的通用底部内容视图类（继承自框架的视图基类 Class_View）

Class_View_Init_User_Info //框架的 User_Info 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Login //框架的 Login 模块视图类（继承自框架的视图基类 Class_View）

Class_View_Memory //框架的 Memory 模块视图类（继承自框架的视图基类 Class_View）

Class_View_Build_Menu //框架的 Build 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Chat_Menu //框架的 Chat 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Cookie_Menu //框架的 Cookie 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Database_Menu //框架的 Database 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Elf_Menu //框架的 Elf 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_File_Menu //框架的 File 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Guide_CommonCommand_Menu //框架的 Guide_CommonCommand 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Guide_Menu //框架的 Guide 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Guide_PenetrationTestCommands_Menu // 框架的 Guide_PenetrationTestCommands 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Memory_Menu //框架的 Memory 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_ProxyShell_Menu //框架的 ProxyShell 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Report_Menu //框架的 Report 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Scan_Menu //框架的 Scan 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Security_Menu //框架的 Security 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Server_Menu //框架的 Server 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Session_Menu //框架的 Session 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Shell_Menu //框架的 Shell 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_User_Menu //框架的 User 模块菜单视图类（继承自框架的视图基类 Class_View）

Class_View_Wget_Menu //框架的 Wget 模块菜单视图类（继承自框架的视图基类 Class_View）

（4）特色功能介绍

Phpsploit-Framework 软件首创了基于原始套接字的 Shell 通信环境（ProxyShell），在进行 CTF 比赛或渗透测试行为时，可绕过 80%以上的防火墙规则防御。由于 Shell 通信环境基于 IPV6 协议进行直接通信，可使基于传输层进行数据包过滤的规则失效。基于 IPV6 协议的扩展头部特性，使用 IPV6 协议进行通信的 Shell 环境，可实现对于 Shell 命令请求的代理转发功能，可突破基于 IP 地址限制的网络防御环境。综合评价，基于网络层的 Shell 通信环境在进行 CTF 比赛或渗透测试行为时，会具有更高等级的隐匿特性。如运用恰当，此基于原始套接字的 Shell 通信环境，可成为进行 CTF 比赛或渗透测试行为时的有效攻坚利器！

Phpsploit-Framework 软件内部已集成了一套基于 C/S 架构的即时通信系统（Chat），任何部署了 Phpsploit-Framework 软件的环境均可运行 Phpsploit-Framework 软件自带的即时通信系统。此即时通信系统的客户端采用 JAVA 语言编写，服务端采用 PHP 语言编写。通信双方采用 TCP 协议进行通信，通信数据已采用较为安全的 RSA 非对称方式进行加密（但应注意，您在正式创建并使用生产环境的项目程序文件之前，应该自行修改分别位于客户端与服务端源码文件中的通信密钥信息。警告：使用原始的 RSA 密钥进行数据加密是并不安全的！这可能会导致通信数据遭到劫持后被成功解密！客户端的 RSA 密钥信息，位于即时通信项目源码目录的 Chat.System.Lib.Security.Rsa 类中。服务端的 RSA 密钥信息，位于 Phpsploit-Framework 软件的 Class_Base_Security_Rsa 类中。）。在公共聊天环境可能遭受社会工程攻击时，使用 Phpsploit-Framework 软件自带的即时通信功能，

也可以成为一种私密的团队协作与攻防配合选择。

Phpsploit-Framework 软件内部集成多种 Shell 环境，与传统的渗透测试 Shell 环境相比，Phpsploit-Framework 软件自带的 Shell 大部分集成了安全通信的功能。WebShell 已采用动态密钥技术对通信数据进行实时加密，每次通信使用的数据密钥均不相同。基于 C/S 架构的 ServerShell 也已使用了动态 TOKEN 令牌与动态密钥技术对通信数据进行加密。ProxyShell 技术同样采用了基于规则的 TOKEN 令牌与动态加密技术来提升 Shell 环境的通信安全性，同时 ProxyShell 更采取白名单机制最大限度地降低了 ProxyShell 被恶意攻击的风险。

Phpsploit-Framework 软件内部自带的数据库管理功能（database），已对传输的数据进行动态加密处理（每次通信使用的加密钥匙均不相同）。可以更大程度上降低通信数据被劫持与恶意攻击的风险。此数据库管理功能模块被细分为查询（query）与更新（exec）两个子模块，根据模块特性的不同，数据库管理功能模块中 SQL 语句的执行结果，将以不同形式体现（例如针对多条查询或更新语句，将以集合形式显示语句对应的执行结果）。

Phpsploit-Framework 软件内部自带的 ELF 格式文件解析功能，提供了对于 ELF 格式的二进制内容进行内容解析的相关支持。Phpsploit-Framework 软件用户可以使用此功能用于二进制内容的安全分析，此功能可自动提取 ELF 格式文件的文件头、程序头表、节头表、程序表、节表数据，并按照指定格式进行相应显示。在进行 CTF 比赛与渗透测试行为时，此功能可为溢出类模糊测试（Fuzz）提供一定的数据分析支持。结合 Phpsploit-Framework 软件提供的共享内存管理功能一起使用，可进一步验证溢出类或远程代码执行类漏洞是否存在。在进行渗透测试行为时，先于恶意黑客之前，及时发现与有效修复自身系统存在的安全漏洞是十分有必要的！综合评价，Phpsploit-Framework 软件自带的 ELF 格式文件解析功能是非常实用且有广泛应用场景的硬核能力。

Phpsploit-Framework 软件内部自带的共享内存管理功能（memory），提供了对于操作系统中共享内存类资源进行相应管理的能力，此共享内存管理功能可发现与访问的共享内存资源数量与运行 Phpsploit-Framework 软件时的权限息息相关。此功能提供了查看、创建、更新、清除共享内存资源的相关能力，可以单独使用，或者与 Phpsploit-Framework 软件自带的 ELF 格式文件解析功能配合使用，用于二进制安全类的模糊测试（Fuzz）。此共享内存管理功能也管理着 Phpsploit-Framework 软件自身绑定的一些共享内存资源。可以说，此功能是 Phpsploit-Framework 软件中非常重要的核心功能之一。

Phpsploit-Framework 软件内部自带的文件在线下载（wget）功能与传统的渗透测试环境中的文件下载功能相比，具备更多优势。首先，Phpsploit-Framework 软件自带的文件在线下载功能可提供下载进度实时显示，有利于 Phpsploit-Framework 软件用户实时观测文件下载进度。同时，使用文件在线下载功能下载的文件成功在服务端保存后，会生成具有显著标识的文件名（文件名称中带有.phpsploit 字样），可以方便在渗透测试与安全审计行为结束后进行相应清理。

Phpsploit-Framework 软件内部自带的文件管理功能（file），实现了简易的文件管理特性。文件管理功能中提供的搜索功能，可以很方便地在指定路径中搜索指定的目录或文件；文件管理功能中提供的浏览功能，可以非常便捷地查看指定路径下的目录内容；文件管理功能中提供的阅读功能，可以使您看到文件内部的真相（对于文件内容的显示，有两种模式可以切换，分别为二进制格式与纯文本格式。对于大体积文件，支持分页浏览，可基本满足对于体积稍大文件的分析需求）；文件管理功能中提供的文件创建功能，可以帮助您在指定路径下创建指定文件，同时为了方便管理（例如，在渗透测试与安全审计行为进行完毕后，进行相应的文件清理），使用 Phpsploit-Framework 软件创建的新文件的文件名称中将包含“.phpsploit”字样。文件管理功能中提供的文件编辑功能，可以使您非常便捷地编辑通过 Phpsploit-Framework 软件创建或上传的相应文件（但请注意，编辑大体积文件可能导致服务器负载增加、网页卡死、浏览器无响应等情况的发生，请您谨慎对待，避免编辑大体积的文件）；文件管理功能中提供的文件上传功能，可以使您非常方便地上传相应文件到服务器空间（但请务必注意！请不要将您自身无法进行有效管控的文件上传到服务器空间中，这可能会给服务器空间造成很大风险），当文件上传成功后，新上传文件的文件名称中将含有“.phpsploit”字样（这样的设定，主要是为了您在渗透测试与安全审计行为结束后，方便地对上传文件进行相应清理）；文件管理功能中提供的文件删除功能，可以快捷地删除 Phpsploit-Framework 软件用户使用 Phpsploit-Framework 软件直接下载、上传、创建的相关文件；文件管理功能中提供的文件清理功能，可以实现对于相关文件的一键清理（例如，带有“.phpsploit”字样的相关文件）。注意，出于安全考虑，Phpsploit-Framework 软件被设计为仅能编辑、删除 Phpsploit-Framework 软件用户使用 Phpsploit-Framework 软件直接下载、上传、创建的相关文件！对于并未使用 Phpsploit-Framework 软件直接下载、上传、创建的相关文件，Phpsploit-Framework 软件将无法成功对其进行编辑、删除等行为。

Phpsploit-Framework 软件内部自带的扫描功能（scan），可以帮助 Phpsploit-Framework 软件用户便捷地开展 Web 站点存活状态检测、主机端口开放状态检测工作。同时，比较具有特色的设计是，在 Phpsploit-Framework 软件内部自带的扫描功能（scan）中，也为安全运维人员提供了非常友好体验与实用功能。通过使用扫描功能（scan）中提供的样本校验功能(（tamperproof），安全运维人员只需分别输入样本目录与采样目录的相应路径后点击“start scan tamperproof”按钮，即可根据样本目录内容对采样目录进行内容匹配度校验（主要匹配的内容包括： 1、样本目录中的子目录数量和文件数量与采样目录中的子目录数量和文件数量是否相同； 2、采样目录及其下属子目录中是否出现样本目录及其对应子目录中并不存在的目录或文件； 3、样本目录及其下属子目录中存在的目录或文件是否在采样目录及其对应子目录中并未发现；4、样本目录及其下属子目录中文件是否与采样目录及其下属子目录中的文件大小不同，以及文件的 md5 校验值或者 sha1 校验值不同等）。同时，样本校验功能(（tamperproof）将对发现的异常内容给予警告提示（通常是红色字样的相关信息）！

Phpsploit-Framework 软件内部自带的加解密功能（security）中，提供了大量的、较为实用的在线加解密工具。Phpsploit-Framework 软件用户，无需再通过各种渠道去搜索对应的数据加密与解密工具！通过使用 Phpsploit-Framework 软件内部

自带的加解密功能（security），即可完成对于绝对大部分通用数据的加解密操作（比较常用的加解密操作，包括 url encode/decode , base64 encode /decode , crypt encode/decode , openssl encode/decode , md5 , sha1 , hash 等）！

Phpsploit-Framework 软件内部自带的报表功能（report），可以非常方便地帮助 Phpsploit-Framework 软件用户在线创建漏洞报告文档（xls 格式）。通过此报表功能（report），Phpsploit-Framework 软件用户能够在未安装 Excel 软件的情况下，也可以十分方便地创建漏洞报告！当漏洞报告撰写完成，Phpsploit-Framework 软件用户可以通过报表功能（report）自带的导出功能（export vulnerability report）在线下载根据漏洞报告内容自动生成的报表文件。

Phpsploit-Framework 软件内部提供的指导手册功能（guide）中，集合了 Phpsploit-Framework 软件作者用时 1 年多时间来不断收集与整理的大量渗透测试/安全运维文档资料（其中对于大量英文资料的相关翻译，更耗费了 Phpsploit-Framework 软件作者海量的精力与时间）。在指导手册功能（guide）中，集成了各类命令的具体用法。包括信息收集、漏洞分析、Web 程序、数据库评估、密码安全、无线安全、逆向工程、漏洞利用、嗅探/欺骗、权限维持、数字取证、安全报告、社会工程等领域的实用工具相关命令用法等。

（5）注意事项

（a）Phpsploit-Framework 软件用户在正式使用 Phpsploit-Framework 软件之前，应认真阅读用户协议与免责声明的全部内容，并遵循用户协议约定内容规范性地在合法合规前提下使用 Phpsploit-Framework 软件！ 使用 Phpsploit-Framework 软件进行任何未经合法授权的渗透测试与安全审计行为，都是被 Phpsploit-Framework 软件作者明确禁止的！ 如果 Phpsploit-Framework 软件用户违反用户协议约定，将 Phpsploit-Framework 软件用于任何非法用途，Phpsploit-Framework 软件作者不承担任何责任！同时，Phpsploit-Framework 软件作者将依法追究 Phpsploit-Framework 软件用户由于违反用户协议而产生的全部法律责任！

（b）Phpsploit-Framework 软件用户在使用 Phpsploit-Framework 软件时，应注意特权账户与普通账户之间的区别（尽管它们之间账户名称相同）！ 特权账户，通常被用于特定场合（例如，在 Phpsploit-Framework 软件初始化安装时，进行普通账户的创建，以及动态密码和 md5_token 令牌的分配）。普通账户，主要用于 Phpsploit-Framework 软件的登陆行为和 Phpsploit-Framework 软件的功能使用行为！因此，作为 Phpsploit-Framework 软件用户，您必须牢记每次 Phpsploit-Framework 软件初始化安装时生成的普通账户密码与 md5_token 令牌信息，否则您可能在 Phpsploit-Framework 软件初始化安装完成后，由于未记住普通账户密码和 md5_token 令牌信息而导致发生无法登陆的情况（如果这种情况发生，您只能在 Phpsploit-Framework 软件所处系统的命令行环境中使用特权账户调用 Phpsploit-Framework 软件的/clear 路由来解决此问题！如果仍然失败，您只能尝试执行 ipcrm -M 0x5d8a0000 和 ipcrm -M 0x5d8a0001 来解决问题！但在

Phpsploit-Framework 软件所处环境中直接操作共享内存资源是存在风险的！ 尽管 Phpsploit-Framework 软件在一般情况下，会使用 KEY 0x5d8a0000 和 KEY 0x5d8a0001 对应的共享内存资源来进行 Phpsploit-Framework 软件普通账户信息的相应存储，但其它软件也有可能会占用这两块共享内存资源（尽管这种情况发生的几率并不高！同时，除非其它软件占用这两块共享内存资源时，对于这两块共享内存资源的权限存在设置不当，导致 Phpsploit-Framework 软件可以对这两块共享内存资源进行成功读写，否则 Phpsploit-Framework 软件在 0x5d8a0000 和 0x5d8a0001 对应的共享内存资源已被占用的情况下，是无法初始化成功的）！ Phpsploit-Framework 软件使用的 KEY 0x5d8a0000 和 KEY 0x5d8a0001 会存在相应特征信息，可作为参考供您酌情进行相应操作。正常情况下，Phpsploit-Framework 软件使用的 KEY 0x5d8a0000 对应的共享内存大小为 32 字节，内存访问权限为 8 进制数 660；Phpsploit-Framework 软件使用的 KEY 0x5d8a0001 对应的共享内存大小为 1048712 字节，内存访问权限为 8 进制数 660。如您没有把握正确地清理相关共享内存资源，您可联系 Phpsploit-Framework 软件所处环境的系统管理员，与您共同解决问题。最坏的结果一般是，您可能需要与 Phpsploit-Framework 软件所处环境的系统管理员一起通过重新启动 Phpsploit-Framework 软件所处环境服务器的方式来解决问题！但这是可以预料的最糟糕情况！在一般正常情况下，清理掉 Phpsploit-Framework 软件使用的 KEY 0x5d8a0000 和 KEY 0x5d8a0001 对应的共享内存资源，即可有效解决问题）！ 在普通账户登陆成功之后，在 Web 环境中，Phpsploit-Framework 软件使用 Session 会话信息进行浏览器与 Web 服务器环境中 Phpsploit-Framework 软件的通信认证。在 Cli 环境中，Phpsploit-Framework 软件使用 Phpsploit-Framework 软件初始化安装时创建的 md5_token 令牌进行通信认证。由于普通账户的认证信息存储使用共享内存资源（Share Memory）作为载体，那么当 Phpsploit-Framework 软件的运行环境并未开启 shmop 扩展时，Phpsploit-Framework 软件将会利用特权账户兼容普通账户使用，此时，特权账户与普通账户的账户名称与账户密码均相同。

（c）Phpsploit-Framework 软件用户在尝试使用 Phpsploit-Framework 软件的 /build 功能构建生产环境的程序项目文件时，请注意，Phpsploit-Framework 软件用户需要把常量 PRIVILEGE_USER_MODULE_USER 和 常量 PRIVILEGE_USER_MODULE_USER 的值，分别设置为生产环境的特权账户名称及对应密码！ Phpsploit-Framework 软件用户可以通过访问路由 /user/create_production_privilege_user_password 来获得正确的生产环境特权账户名称及对应密码，此路由被设置为专用于创建正确的生产环境特权账户名称及对应密码信息。

（d）Phpsploit-Framework 软件用户在正式创建并使用生产环境的项目程序文件之前，应该自行同步修改位于即时通信系统中客户端（java）源码文件中的通信密钥信息与位于 Phpsploit-Framework 软件中（php）源码文件中的通信密钥信息！注意，使用客户端的 RSA 公钥加密的数据，需要使用客户端的 RSA 私钥才能解密成功！使用服务端的 RSA 公钥加密的数据，需要使用服务端的 RSA 私钥才能解密成功！提示，客户端与服务端的相应源码中均存储了客户端与服务端的 RSA 公私钥信息，您需要对它们进行同步更新，以保持 RSA 密钥信息一致！警告：使用原始的 RSA 密钥进行数据加密是并不安全的！这可能会导致通信数据遭到劫持后被成功解密！客户端的 RSA 密钥信息，位于即时通信系统的项目源码目录的 Chat.System.Lib.Security.Rsa 类中。服务端的 RSA 密钥信息，位于 Phpsploit-Framework

软件的 Class_Base_Security_Rsa 类中。

（e）Phpsploit-Framework 软件用户在使用 Phpsploit-Framework 软件时，应尽量选择较为安全的网络环境！在基于 Web 类应用的应用层协议中，HTTPS 协议远比 HTTP 协议更加安全！位于 TCP 与 HTTP 协议中间的 SSL 层，可以最大限度保障通信数据的安全！但是，您仍需警惕来自虚假网关的中间人欺骗攻击！

（f）此技术白皮书文件中部分内容采用翻译软件自动生成，并非 100%使用人工方式书写，因此可能会存在一些内容错误！如您发现此文件内容中存在相关错误，欢迎联系 Phpsploit-Framework 软件作者对相关内容进行检查及修正，不胜感激!