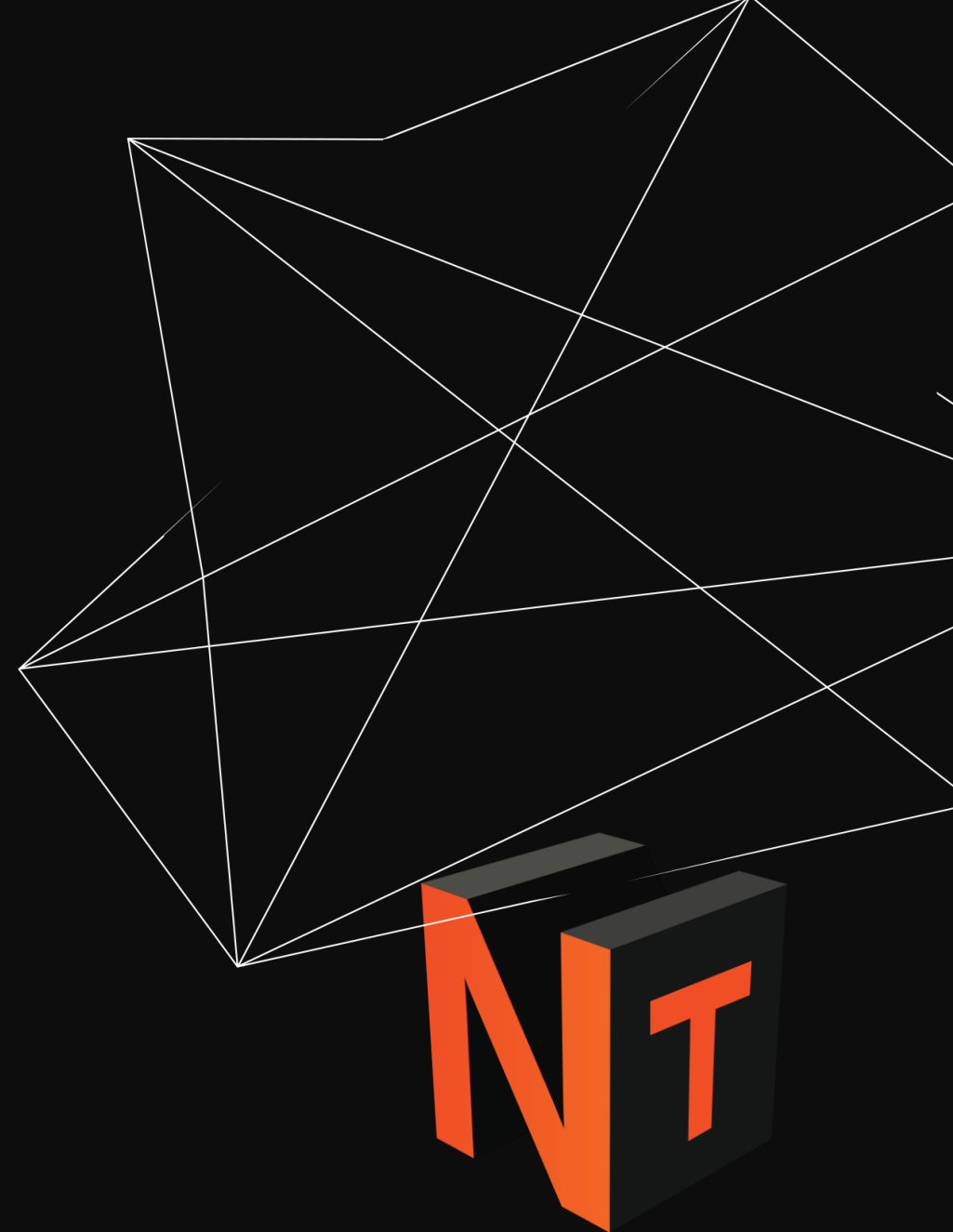


Ko se heker in razvijalec srečata na virtualni kavi

Milan Gabor
Gregor Spagnolo



Heker

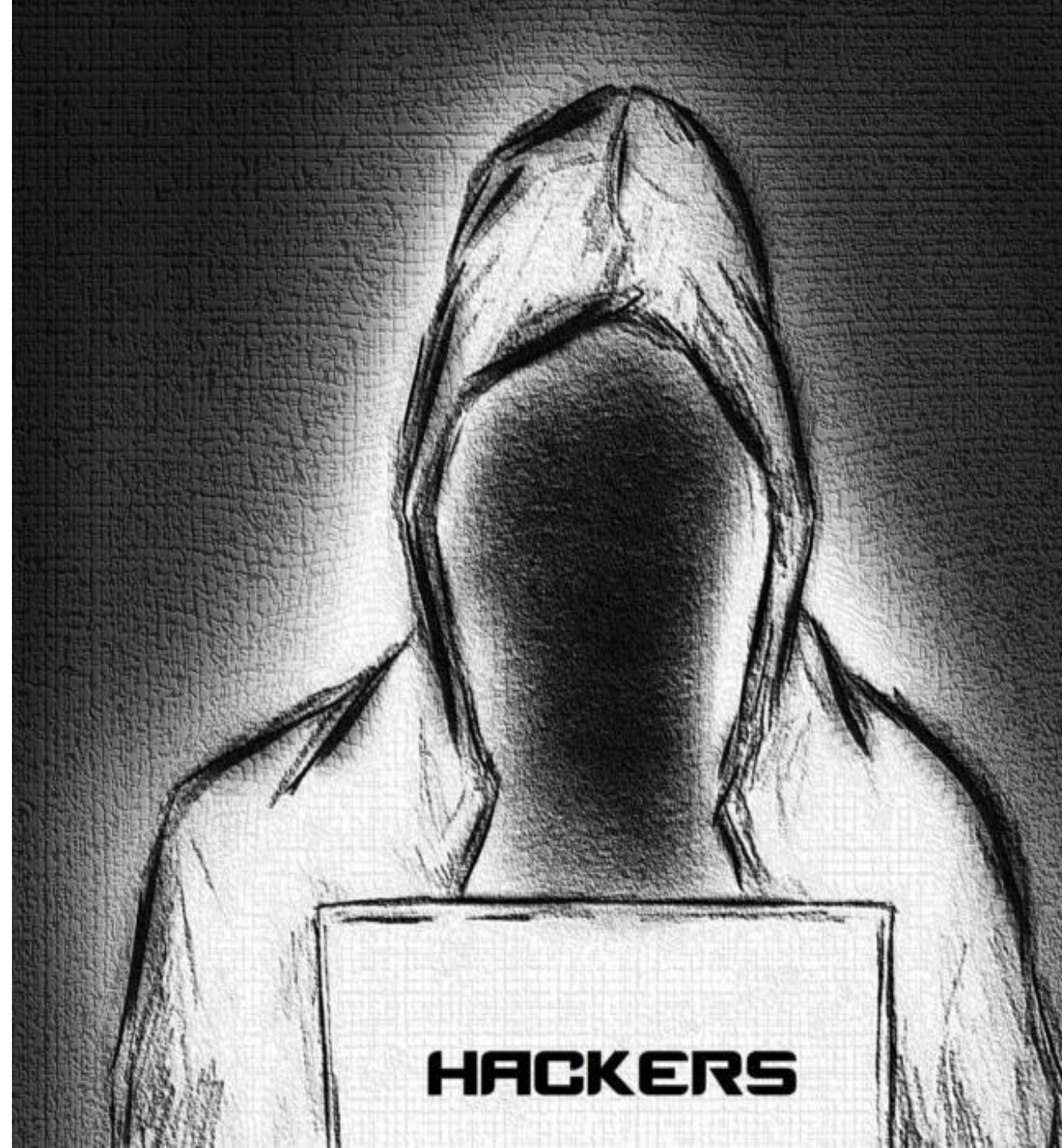
Ni razvijalec

Rad razbija stvari

Posebej rad ima nevešče razvijalce



#ntk18



Orodja

Kali Linux



#ntk18



DUAL BOOT KALI LINUX 2018.1 WITH WINDOWS 10



Razvijalec

G33k

Latest&Gratest

Ni še bil na izobraževanju s tega
področja



#ntk18



Orodja

Visual studio

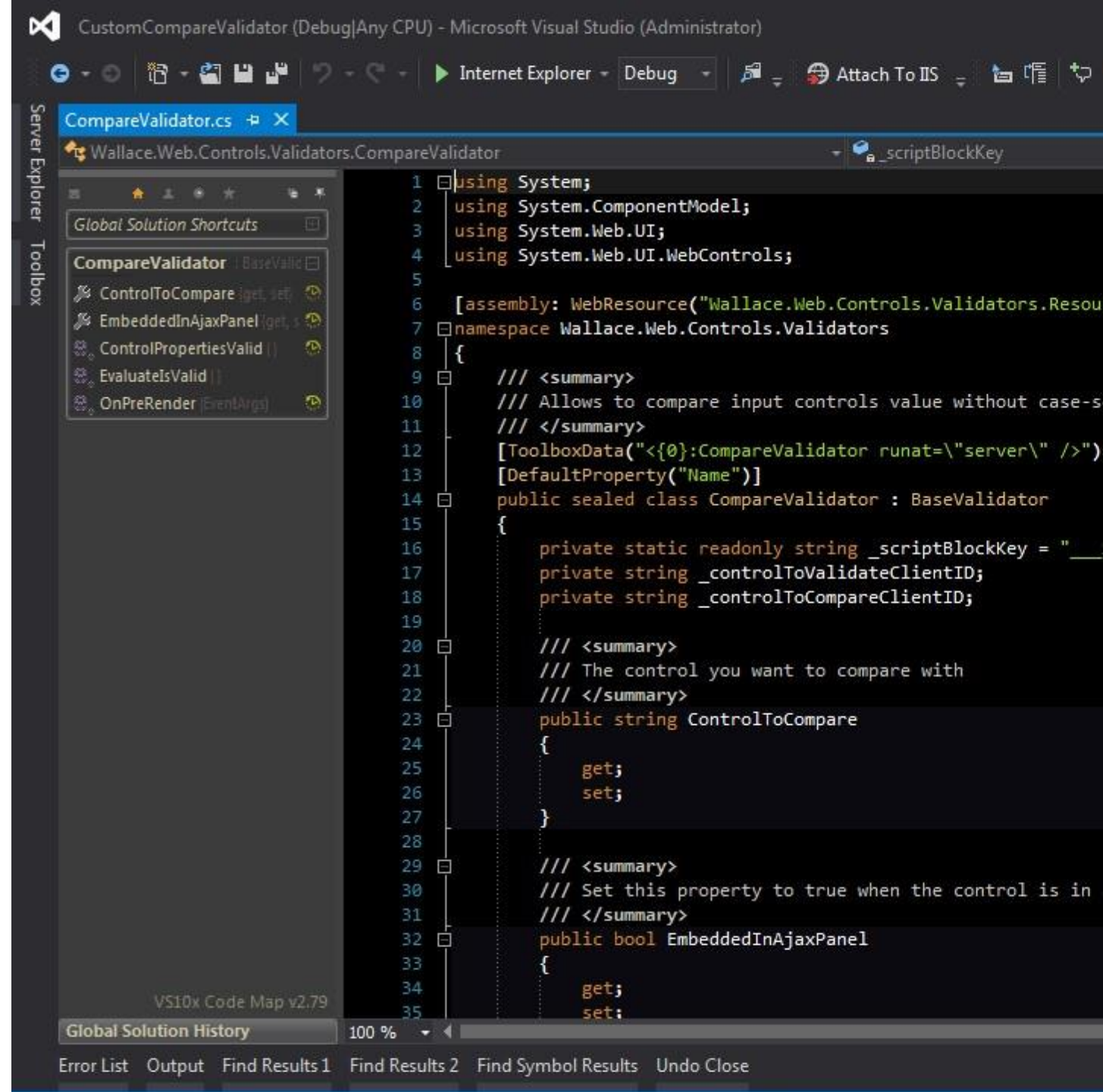
Git

Docker

Linux



#ntk18



Aplikacije

Velika množica pregledanih aplikacij

Veliko ponavljajočih se napak

Niso odvisne od tehnologije

Copy paste

Roki



#ntk18



O aplikaciji

.Net Core

Jquery

API

Docker build

Git SC



C#/.NET App

60 in
Minutes



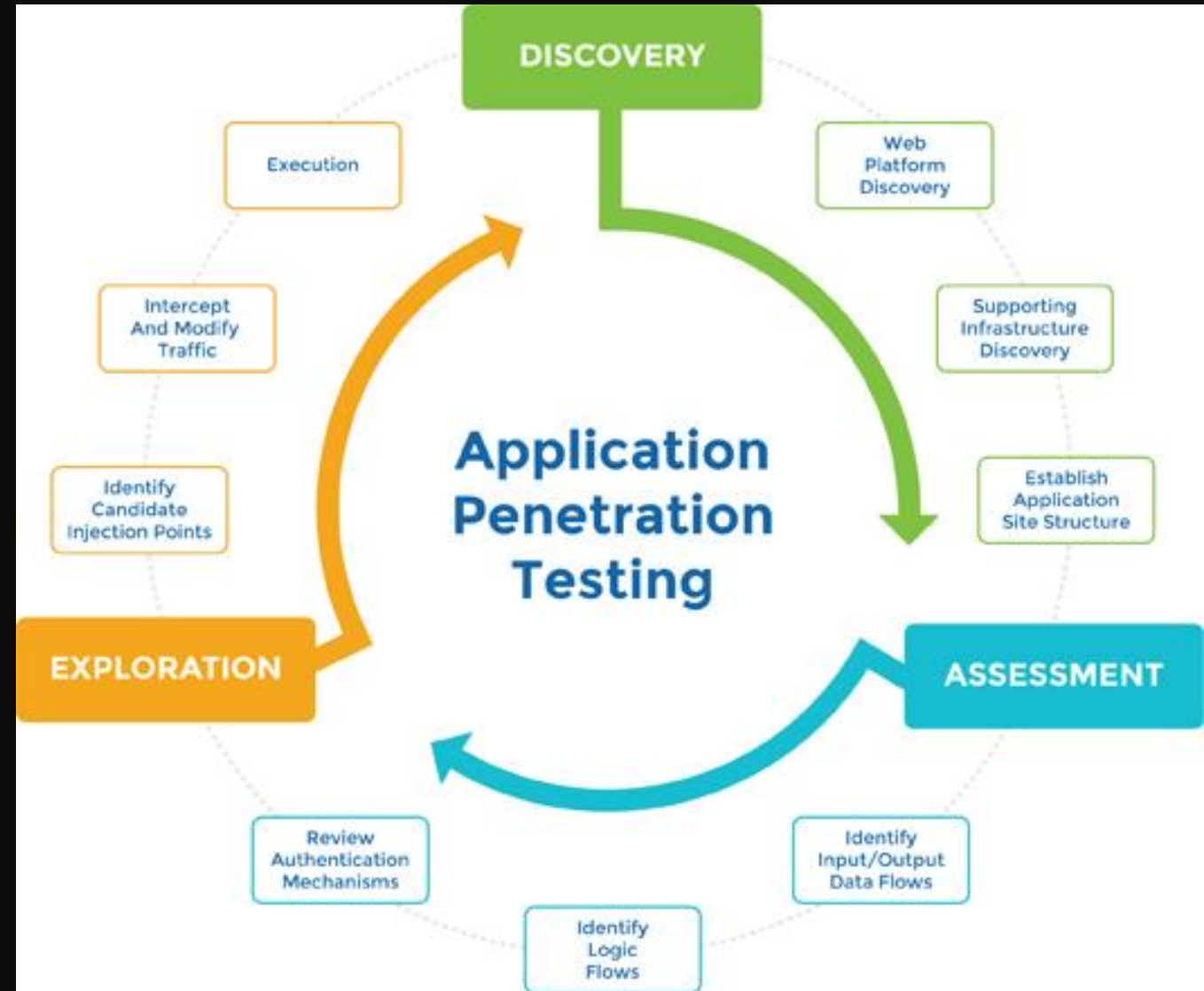
#ntk18



Virtual
Coffee
Date

Faze

1. OSINT
2. Avtomatska analiza
3. Analiza in identifikacija
4. Kraja podatkov, razmaličenje, ...
5. Izkoriščanje



OSINT

Google

Shodan

Drugi iskalniki



#ntk18



Aktivno skeniranje

Nikto

OWASP ZAP

Burp

Drugi namenski skenerji in proksiji



#ntk18

Aktivno izkoriščanje

OWASP TOP 10

OWASP Testing metodologija

Druge aktivnosti na podlagi izkušenj

Včasih nujne nore ideje



#ntk18

Najdene stvari

- git
- swagger
- auth controller logging passwords
- APITransaction -> brez autorizacije
- Logs (exposed)
- transaction create stored XSS
- transaction create senderId se lahko spreminja
- search sql injection



OWASP

- Open Web Application Security Project
 - Odprta skupnost
 - Neprofitna organizacija
- Glavni cilj
 - Varna programska oprema
- <https://www.owasp.org>



OWASP TOP 10

OWASP Top 10 - 2013	➔	OWASP Top 10 - 2017
A1 – Injection	➔	A1:2017-Injection
A2 – Broken Authentication and Session Management	➔	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	➔	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	➔	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	➔	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	➔	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]



Primer: OWASP Security Shepherd

Open and Close Modules

Open or Close by Category

View Feedback

View Progress

User Management

Scoreboard

Field Training

Private

- X Insecure Data Storage
- ✓ Insecure Cryptographic Storage
- X SQL Injection
- X Insecure Cryptographic Storage

Challenge 1

- X Insecure Direct Object Reference

Challenge 1

- X Poor Data Validation 1
- X SQL Injection 1

Corporal

Sergeant

Lieutenant

Major

The following form's parameters are concatenated to a string that will be passed to a SQL server. This means that the data can be interpreted as part of the code.

The objective here is to modify the result of the query with **SQL Injection** so that all of the table's rows are returned. This means you want to change the **boolean** result of the query's **WHERE** clause to return true for every row in the table. The easiest way to ensure the **boolean** result is always true is to inject a **boolean 'OR'** operator followed by a true statement like `1 = 1`.

If the parameter is been interpreted as a string, you can escape the string with an apostrophe. That means that everything after the apostrophe will be interpreted as SQL code.

Hide Lesson Introduction

Exploit the **SQL Injection** flaw in the following example to retrieve all of the rows in the table. The lesson's solution key will be found in one of these rows! The results will be posted beneath the search form.

Please enter the **user name** of the user that you want to look up

Get this user

Search Results

User Id	User Name	Comment
12345	user	Try Adding some SQL Code
12346	OR 1 = 1	Your Close, You need to escape the string with an apostrophe so that your code is interpreted
12543	Fred Mtenzi	A lecturer in DIT Kevin Street
14232	Mark Denihan	This guy wrote this application
61523	Cloud	Has a Big Sword
82642	qwldshs@ab	Lesson Completed. The result key is 3c17f6bf34080979e0cebda5672e989c07ceec9fa4ee7b7c17c9e3ce26bc63e0



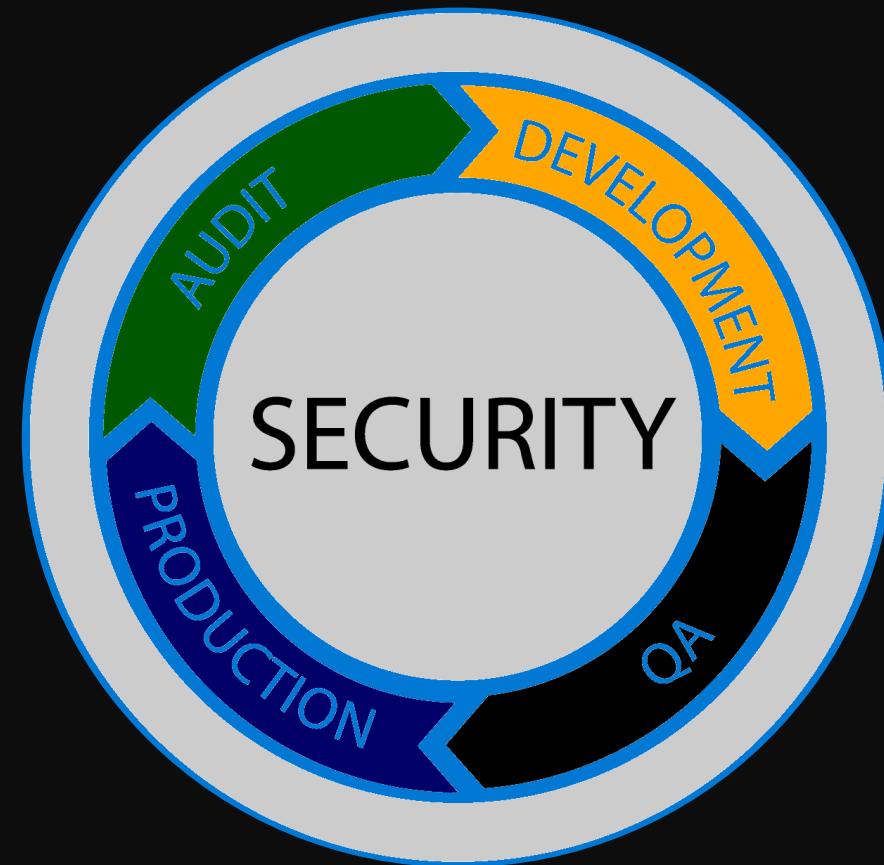
Priporočila za varnejše spletne Aplikacije

- Sistem
 - Popravki
 - Ustrezne verzije
 - Nadzor
 - Odstranitev nepotrebnih aplikacij
- Aplikacije
 - Zadnja verzija?
 - Nepotrebni moduli?
 - Izveden pregled ali vsaj avtomatski sken?



Kako zaščiti aplikacije/sisteme

- Vgraditi varnost v cel življenjski cikel
- Uporabiti varnostne principe v vseh fazah in aktivnostih
- Izobraževanje
 - Zavedanje se potencialnih napadov
 - Pošiljanje ljudi na različne izobraževanja (tudi NTK šteje)



Izobraževati

- **Razvijalce** – Najboljše prakse iz varnega razvoja SW
- **Testerje** – Metode za identifikacijo ranljivosti
- **Varnostne inženirje** – Proces varnega razvoja SW
- **Lastnike, menedžment** – Razumeti tveganja in zakaj je to sploh pomembno za varnost spletnih aplikacij

Ustrezna kombinacija

Sharing is caring

Skupaj za istim ciljem

Together
WE
Achieve
More



#ntk18

{ THE **SECURE** } DEVELOPER }

Varno programiranje

Seznanitev s tipičnimi napakami

Uporaba zadnjih tehnologij

Za noobs in seniors

Kombinacija napada in obrambe

Developer + Hacker



#ntk18



thank you

děkuji gracie sobodi fiik akpé tñnan mēsi dankie tahg spas gracias paldies agat kié eskerrik arigatō terima edirem

mercici choukrane aciuy milesker dahn-kah faleminderit hvala fyri takk

trugéré bo chai dhanyavadagalu murakoze dank gratias kyay dhanyabaad shoo-kree-a

blagodaria chnorakaloutioun syeh-syeh hamnida shoe-krahn wel aguyjé ago tin

deh-ku-yih tzu tanemirt tesekkur trugarez daw-dyeh aabhar agimus gracies mahalo kamsah Barak much

khob raibh olun kee-toas toh-dah kiitos dahn-kee dankewol danke çox toda asko Allahu maith

dankon