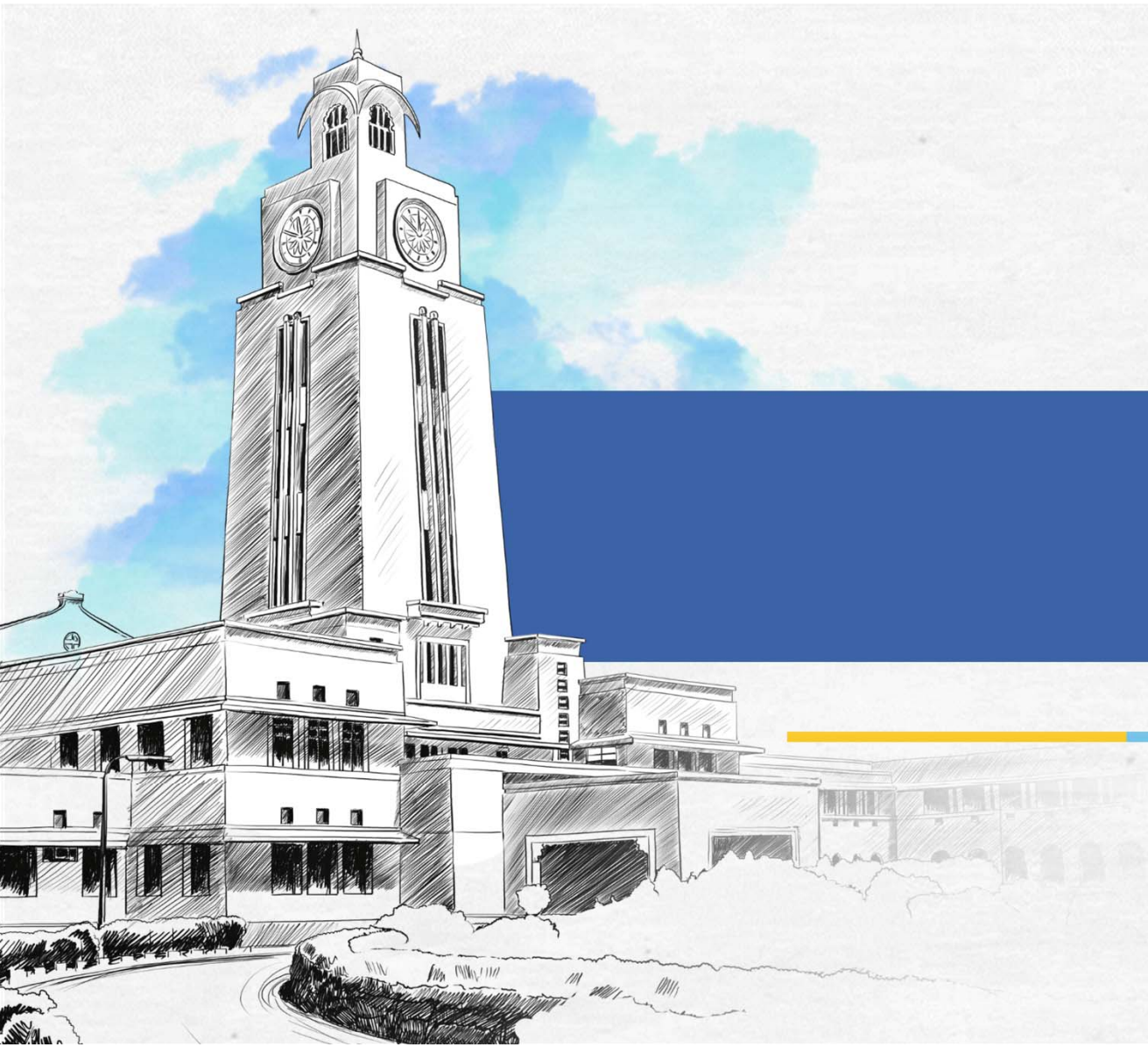


BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Cyber Security

Common Cyber Attacks

Dr. Ramakrishna Dantu
Associate Professor, BITS Pilani

Disclaimer and Acknowledgement

- The content for these slides has been obtained from books and various other source on the Internet
- I here by acknowledge all the contributors for their material and inputs.
- I have provided source information wherever necessary
- I have added and modified the content to suit the requirements of the course

Common Cyber Attacks

Agenda

- Common Cyber Attacks – Practical Strategies for Identification, Containment and Mitigation:
  - Malware Attacks
    - E.g., Ransomware Attacks
  - Denial of Service Attacks
  - Session Hijacking and Man-in-the-Middle Attacks
  - Phishing and Spear Phishing Attacks
  - SQL Injection Attacks
  - Zero Day Exploits
  - DNS Tunneling Attacks

Common Cyber Attacks

innovate achieve lead

Types of Attacks

- Software Attacks
  - Malware
    - Adware
    - Virus
      - Boot virus
      - Macro virus
      - Memory-resident virus
      - Non-memory-resident virus
    - Polymorphic Threats
    - Spyware
    - Trojan horses
    - Worms
    - Virus and Worm Hoaxes
    - Zero-day attack
  - Back Doors
    - Maintenance hook
    - Trap door
  - Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks
  - Email Attacks
    - Mail Bomb
    - Spam
  - Communications Interception Attacks
    - Packet Sniffer
    - Spoofing
    - Pharming
    - Man-in-the-Middle
    - Domain Name System (DNS) cache poisoning or DNS spoofing
    - Session hijacking or TCP hijacking.

Source: Principles of Information Security by Whitman and Mattord

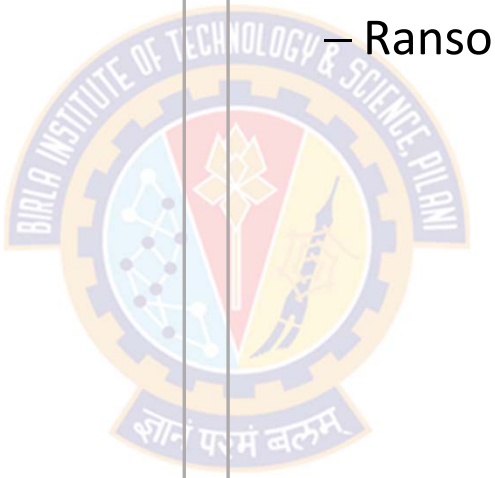
Common Cyber Attacks

innovate achieve lead

Types of Attacks

- Espionage or Trespass
  - Password Attacks
    - Brute Force
    - Dictionary Attacks
    - Rainbow Tables
    - Social Engineering
- Human Error or Failure
  - Social Engineering
    - Advance-fee fraud (AFF)
    - Phishing
    - Pretexting
    - Spear phishing

- Information Extortion
  - Ransomware

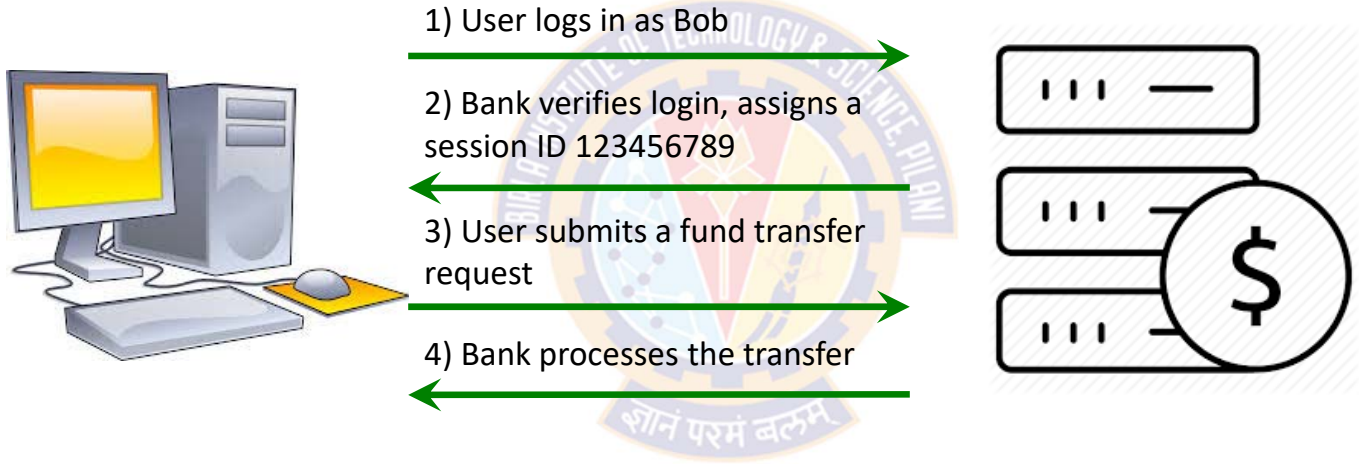


Source: Principles of Information Security by Whitman and Mattord

Session Hijacking

Session Hijacking

How a Session Works



Source: https://securityboulevard.com/2020/11/the-ultimate-guide-to-session-hijacking-aka-cookie-hijacking/

Session Hijacking

innovate achieve lead

Overview

- What is session hijacking?
- How is the session maintained?
- How to perform session hijacking?
- What can attackers do after successful session hijacking?
- Methods used in session hijacking:
  - Cross-site scripting (XSS)
  - Session side jacking
  - Session fixation
  - Cookie theft by malware or direct access
  - Brute force

Source: https://www.netsparker.com/blog/web-security/session-hijacking/

Session Hijacking

innovate achieve lead

Methods used in Session Hijacking

- Cross-site scripting (XSS)

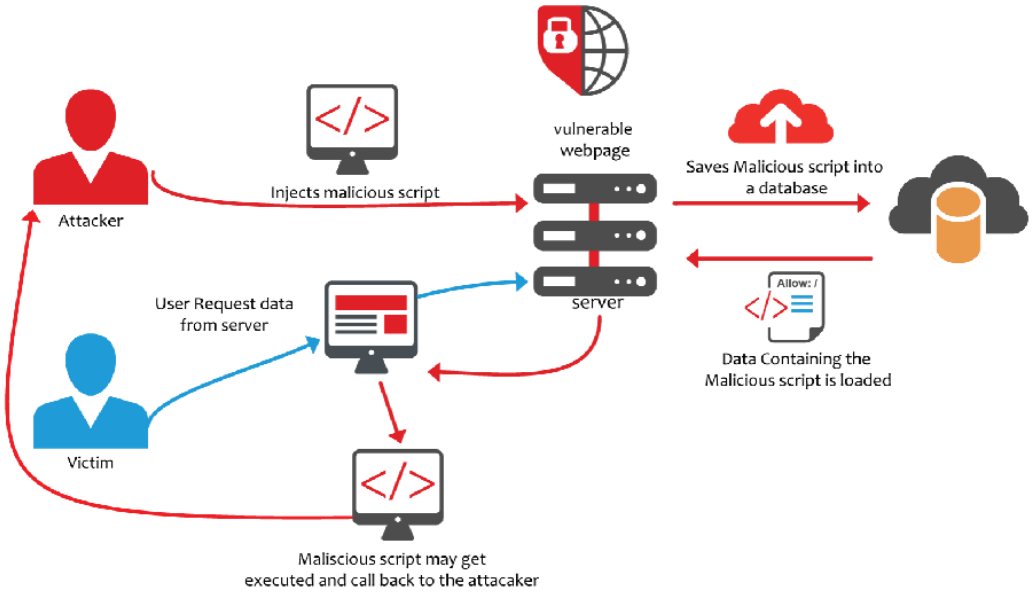
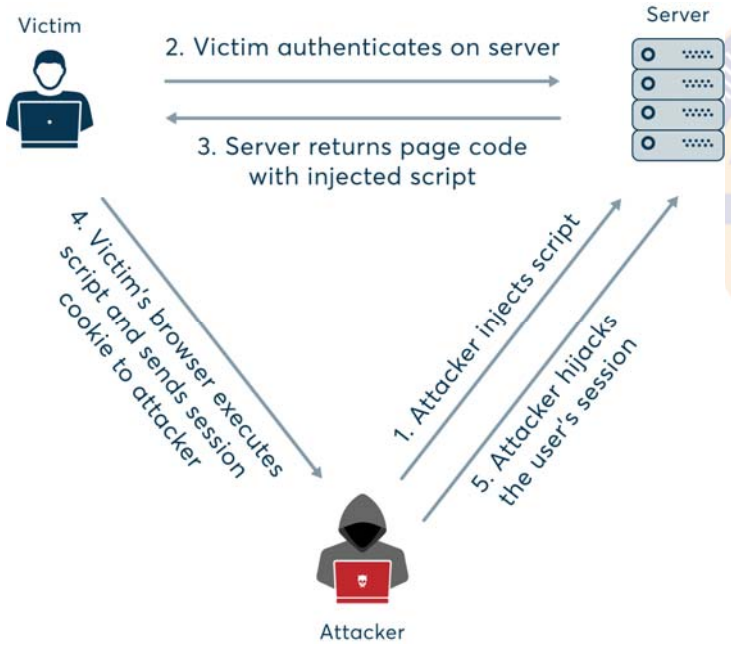


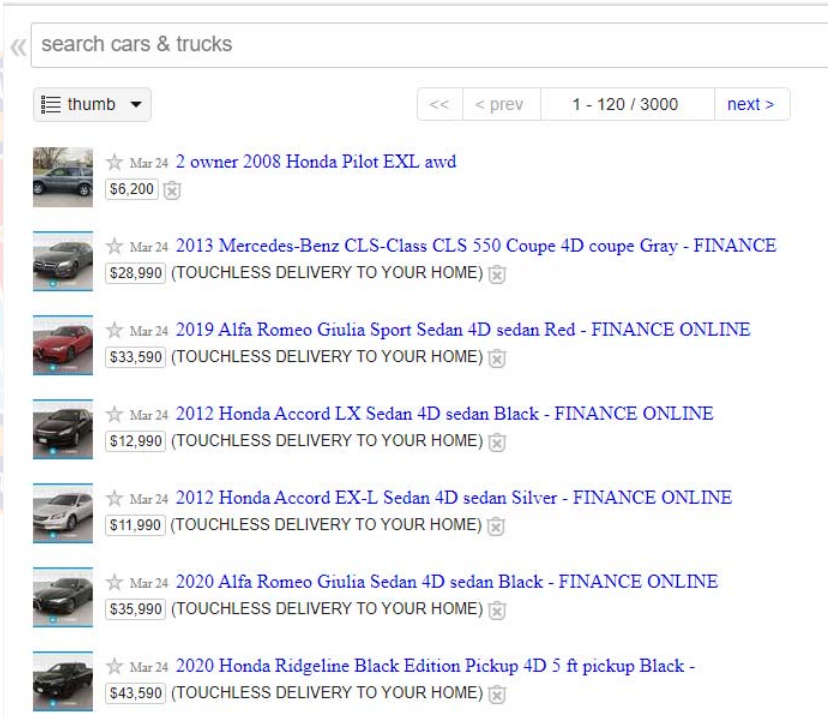
Image Source: https://blogs.sap.com/2015/12/17/xss-cross-site-scripting-overview-with-contexts/

Session Hijacking

Methods used in Session Hijacking

- Cross-site scripting (XSS)



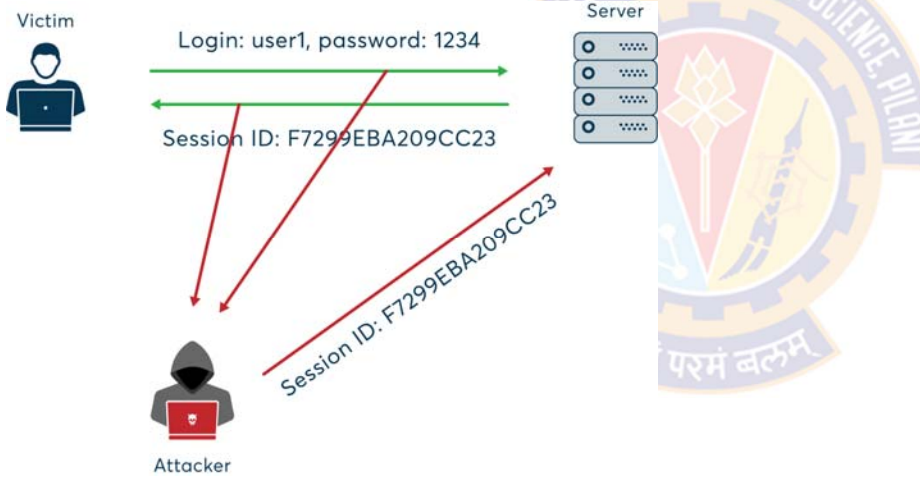


Source: YouTube - XSS - Cross Site Scripting Explained

Session Hijacking

Methods used in Session Hijacking

- Session Side Jacking



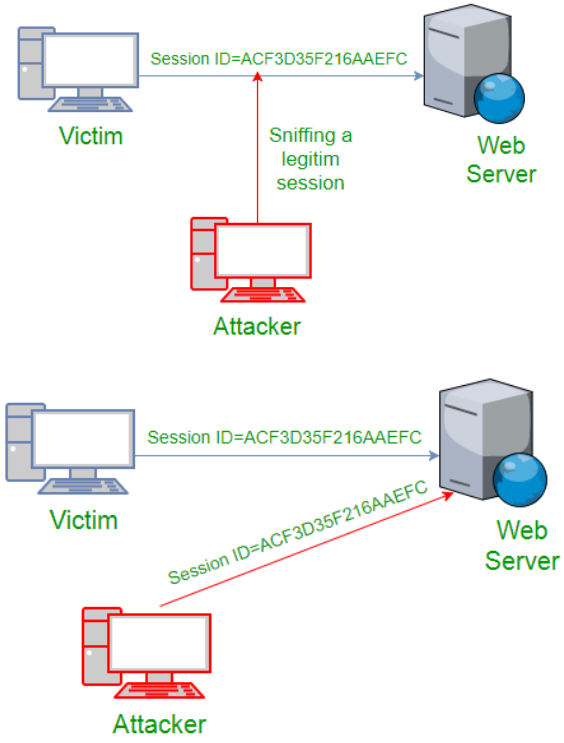


Image Source: https://securityboulevard.com/2020/11/the-ultimate-guide-to-session-hijacking-aka-cookie-hijacking/
Image Source: https://www.geeksforgeeks.org/session-hijacking/

Session Hijacking

innovate achieve lead

Methods used in Session Hijacking

- Session Fixation

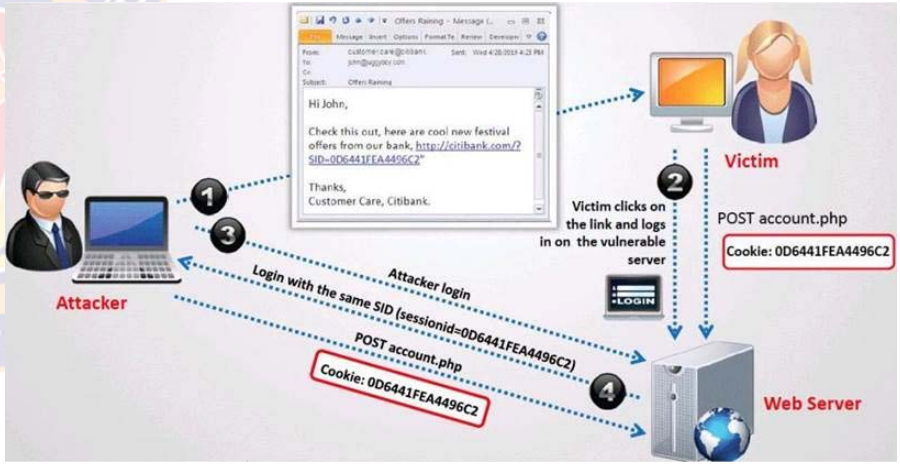


Image Source: https://securityboulevard.com/2020/11/the-ultimate-guide-to-session-hijacking-aka-cookie-hijacking/
Image Source: https://www.exam4training.com/

Session Hijacking

Methods used in Session Hijacking

- Cookie theft by malware
  - Involves installing malware on the user's machine to perform automated session sniffing
  - Once installed, the malware scans the user's network traffic for session cookies and sends them to the attacker
- Cookie theft by direct access
  - Another way of obtaining the session key is to directly access the cookie file in the client browser's temporary local storage (often called the cookie jar)
    - This task can be performed by malware, but also by an attacker with local or remote access to the system.

Source: https://www.netsparker.com/blog/web-security/session-hijacking/

Session Hijacking

Methods used in Session Hijacking

- Brute force
  - The attacker can simply try to guess the session key of a user's active session, which is feasible only if the application uses short or predictable session identifiers
  - In the distant past, sequential keys were a typical weak point, but with modern applications and protocol versions, session IDs are long and generated randomly
  - To ensure resistance to brute force attacks, the key generation algorithm must give truly unpredictable values to make guessing attacks impractical

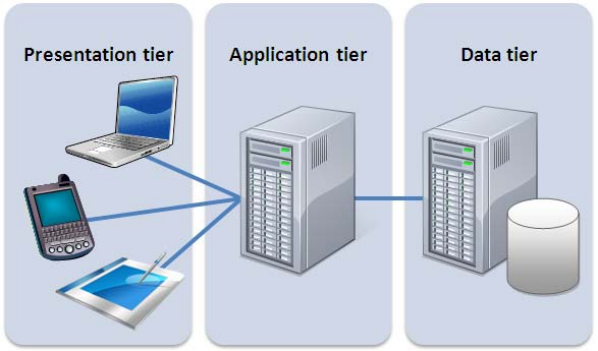
Source: https://www.netsparker.com/blog/web-security/session-hijacking/

SQL Injection

SQL Injection

Overview

- An SQLi attack is designed to exploit the nature of Web application pages
- In contrast to the static Web pages, current Web sites have dynamic components and content
- Dynamic web pages accept data such as location, personal identity information, and credit card information
- This dynamic content is usually transferred to and from back-end databases
- These databases contain all kinds of data
  - For e.g., cardholder data, product data, customer data, purchase information etc.,.
- An application server makes SQL queries to databases to send and receive information



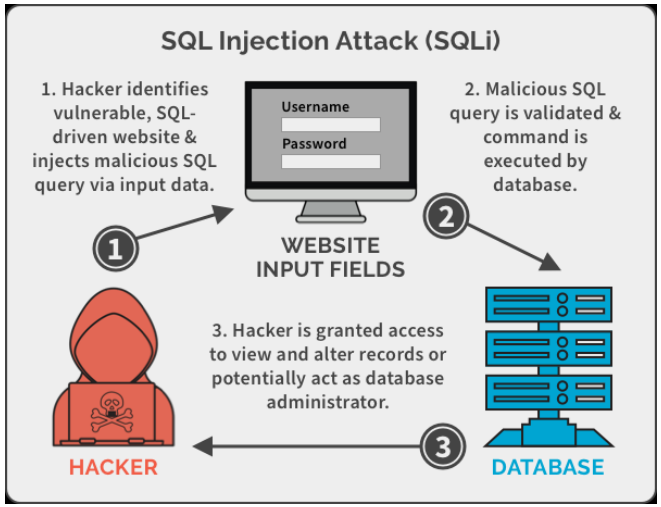
Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

SQL Injection

innovate achieve lead

Overview

- An SQLi attack involves sending malicious SQL commands to the database server
- The most common goal of this attack is bulk extraction of data
- SQL injection can also be exploited to:
  - modify or delete data,
  - execute arbitrary operating system commands, or
  - launch denial-of-service (DoS) attacks
- The attack is viable when user input is either
  - incorrectly filtered for string literal escape characters embedded in SQL statements or
  - user input is not strongly typed, and thereby unexpectedly executed



Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

SQL Injection

innovate achieve lead

Steps involved in a typical SQLi attack

1) Hacker finds a vulnerability in a custom Web application and injects an SQL command to a database by sending the command to the Web server
   - The command is injected into traffic that will be accepted by the firewall
2) The Web server receives the malicious code and sends it to the Web application server
3) The Web application server receives the malicious code from the Web server and sends it to the database server
4) The database server executes the malicious code on the database
   - The database returns data
5) The Web application server dynamically generates a page with data from the database
6) The Web server sends the details over to the hacker

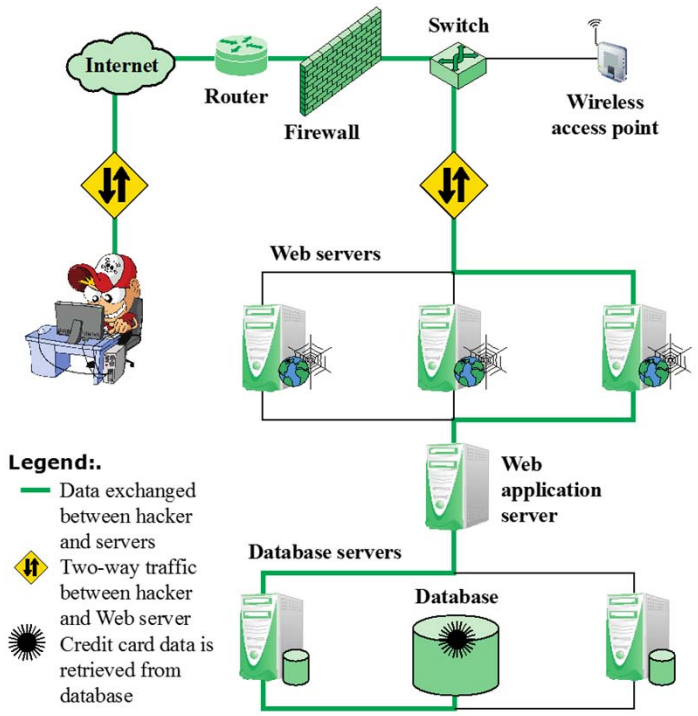


Figure 5.5 Typical SQL Injection Attack

Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

SQL Injection

The Injection Technique

- Consider a script that build an SQL query by combining predefined strings with text entered by a user:

```
var Shipcity;
ShipCity = Request.form ("ShipCity");
var sql = "select * from OrdersTable where ShipCity = '" + ShipCity + "'";
```

- The intention of the script's designer is that a user will enter the name of a city
- For example, if the user enters Redmond, then the following SQL query is generated:

```
SELECT * FROM OrdersTable WHERE ShipCity = 'Redmond'
```

SQL Injection

The Injection Technique

- Suppose that the user enters the following:

```
Redmond'; DROP table OrdersTable--
```

- This results in the following SQL query:

```
SELECT * FROM OrdersTable WHERE ShipCity = 'Redmond'; DROP table
OrdersTable--
```

- The semicolon separates two commands, and the double dash indicates that the remaining text of the current line is a comment and not to be executed
- When the SQL server processes this statement, it will first
  - select all records in OrdersTable where ShipCity is Redmond, then
  - executes the DROP request, which deletes the table

SQL Injection

SQLi Attack Avenues and Types

- SQLi attacks can be characterized in terms of the avenue of attack and the type of attack
- The main attack avenues are:
  - User input
  - Server variables
  - Second-order injection
  - Cookies
  - Physical user input
- Attack types can be categorized as:
  - in-band
  - inferential, and
  - out-of-band

SQL Injection

SQLi Attack Types

- Inband Attack Types
  - Tautology
  - End-of-line comment
  - Piggybacked queries
- Inferential Attack Types
  - Illegal/logically incorrect queries
  - Blind Sql injection

- Uses the same communication channel for injecting SQL code and retrieving results
- The retrieved data are presented directly in the application Web page

- There is no actual transfer of data, but the attacker is able to reconstruct the information by sending particular requests and observing the resulting behavior of the Website/database server

Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

SQL Injection

SQLi – The Main Attack Avenues

- User input
  - Attackers inject SQL commands by providing suitably crafted user input
  - A Web application can read user input in several ways based on the environment in which the application is deployed
  - In most SQLi attacks that target Web applications, user input typically comes from form submissions that are sent to the Web application via HTTP GET or POST requests
  - Web applications are generally able to access the user input contained in these requests as they would access any other variable in the environment

SQL Injection

SQLi – The Main Attack Avenues

- Server variables
  - Server variables are a collection of variables that contain HTTP headers, network protocol headers, and environmental variables
    - https://www.w3schools.com/asp/coll_servervariables.asp
  - Web applications use these server variables in a variety of ways
    - E.g., logging usage statistics, Identifying browsing trends, etc.,.
  - If these variables are logged to a database without sanitization, this could create an SQL injection vulnerability
  - Attackers can exploit this vulnerability by placing data directly into HTTP and network headers
    - https://resources.infosecinstitute.com/topic/sql-injection-http-headers/
  - When the query to log the server variable is issued to the database, the attack in the forged header is then triggered

SQL Injection

innovate achieve lead

SQLi – The main attack avenues

- Second-order injection
  - In the second order injection, a malicious user could rely on data already present in the database to trigger an SQL injection attack
  - The input that modifies the query to cause an attack does not come from the user, but from within the system itself
  - Imagine there is a user already existing in the database with username "dantu"
  - I create another account in the database with username "dantu'--"

A query written like this…

```
UPDATE users
SET password='123'
WHERE username='dantu'--' and password='abc'
```

Translates to…

```
UPDATE users
SET password='123'
WHERE username='dantu'
```

https://haiderm.com/second-order-sql-injection-explained-with-example/

Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

SQL Injection

innovate achieve lead

SQLi – The main attack avenues

- Cookies
  - When a client returns to a Web application, cookies can be used to restore the client's state information
  - An attacker could alter cookies such that when the application server builds an SQL query based on the cookie's content, the structure and function of the query is modified.
- Physical user input
  - SQL injection is possible by supplying user input that constructs an attack outside the realm of web requests
  - This user-input could take the form of conventional barcodes, RFID tags, or even paper forms which are scanned using optical character recognition and passed to a database management system

SQL Injection

SQLi Inband Attack Types

- In-band SQL injection occurs when an attacker is able to use the same communication channel to both launch the attack and gather results
- The retrieved data are presented directly in the application Web page
- In-band SQLi's are typically HTTP GET, POST replies where information retrieved from database is displayed
- With some carefully crafted SQL queries it becomes possible to:
  - enumerate the databases, respective tables, table columns and finally the rows of stored database entries
- The example that we saw earlier was an in-band attack since the same channel was used to launch the attack and obtain the result
  - In this case, it was selecting data from orders table

SQL Injection

SQLi Inband Attack Types

- Tautology
  - Injects code in one or more conditional statements so that they always evaluate to true
  - For example, consider this script, whose intent is to require the user to enter a valid name and password:

```
$query = "SELECT info FROM user WHERE name ='$_GET["name"]' AND pwd =
'$_GET["pwd"]'";
```

  - Suppose the attacker submits "' OR 1=1 --" for the name field
  - The resulting query would look like this:

```
SELECT info FROM users WHERE name = ' ' OR 1=1 -- AND pwpd = ' '
```

  - The injected code effectively disables the password check (because of the comment indicator --) and turns the entire WHERE clause into a tautology
  - Because the conditional is a tautology, the query evaluates to true for each row in the table and returns all of them

SQL Injection

innovate achieve lead

SQLi Inband Attack Types

- End-of-line comment
  - After injecting code into a particular field, legitimate code that follows are nullified through usage of end of line comments
  - By adding "- -" after inputs makes the remaining queries as comments instead of executable code
  - The preceding tautology example is also of this form
- Piggybacked queries
  - The attacker adds additional queries beyond the intended query, piggy-backing the attack on top of a legitimate request
  - This technique relies on server configurations that allow several different queries within a single string of code
  - `SELECT * FROM OrdersTable WHERE ShipCity = 'Redmond'; DROP table OrdersTable`

Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

SQL Injection

SQLi Inferential Attack Types

- Illegal/logically incorrect queries:
  - In an inferential SQLi attack, no data is actually transferred via the web application and the attacker would not be able to see the result of an attack in-band
    - Such attacks are commonly referred to as "blind SQL injection attacks"
  - Instead, an attacker is able to reconstruct the database structure by sending payloads, observing the web application's response and the resulting behavior of the database server.
  - An attacker may gain knowledge by injecting illegal/logically incorrect requests
    - E.g., injectable parameters, data types, names of tables, etc.
  - This attack lets an attacker gather important information about the type and structure of the backend database of a Web application

SQL Injection

SQLi Inferential Attack Types

- In an inferential SQLi attack, no data is actually transferred via the web application
- The attacker would not be able to see the result of an attack in-band
- Here, an attacker is able to reconstruct the database structure by sending payloads, observing the web application's response and the resulting behavior of the database server
- Examples of Inferential Attacks
  - Illegal/Logically Incorrect Queries
  - Blind SQL Injection

SQL Injection

SQLi Inferential Attack Types

- Illegal/logically incorrect queries:
  - An attacker may gain knowledge by injecting illegal/logically incorrect requests
    - E.g., injectable parameters, data types, names of tables, etc.
  - This attack lets an attacker gather important information about the type and structure of the backend database of a Web application
  - The attack is considered a preliminary, information-gathering step for other attacks
  - The vulnerability leveraged by this attack is that the default error page returned by application servers is often overly descriptive
  - The error messages generated can often reveal vulnerable/injectable parameters to an attacker

SQL Injection

SQLi Inferential Attack Types

- Boolean-Based (Content-Based) Blind SQL injection
  - This SQL injection technique relies on sending a SQL query to the database which forces the application to return a different result depending on whether the query returns a TRUE or FALSE result
  - Blind SQL injection allows attackers to infer the data present in a database system even when the system is sufficiently secure to not display any erroneous information back to the attacker
  - The attacker asks the server true/false questions
  - If the injected statement evaluates to true, the site continues to function normally
  - If the statement evaluates to false, although there is no descriptive error message, the page differs significantly from the normally functioning page

https://www.youtube.com/watch?v=1Qs195_8hNw
https://www.youtube.com/watch?v=kRgAtnlFtTo

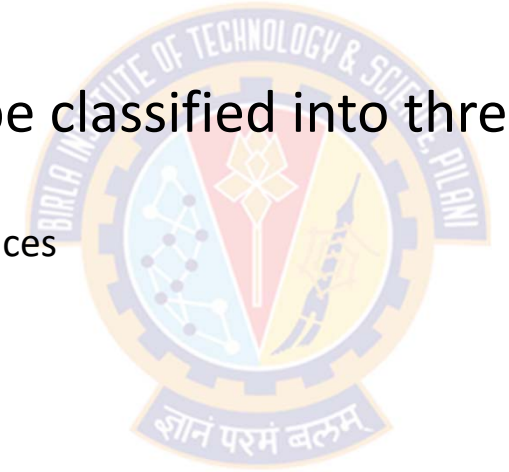
Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

SQL Injection

innovate achieve lead

Countermeasures

- Many SQLi attacks succeed because developers have used insecure coding practices
- The countermeasures can be classified into three types:
  - defensive coding
    - Manual defensive coding practices
    - Parameterized query insertion
    - SQL DOM
  - detection
    - Signature based
    - Anomaly based
    - Code analysis
  - run-time prevention



Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

SQL Injection

Countermeasures – Defensive Coding

- Manual defensive coding practices
  - A common vulnerability exploited by SQLi attacks is insufficient input validation
  - The simple solution for eliminating these vulnerabilities is to apply suitable input validation
  - For example:
    - Input type checking - to check that inputs that are supposed to be numeric contain no characters other than digits
  - This technique can avoid attacks that are based on forcing errors in the database management system
  - Another type of coding practice is one that performs pattern matching to try to distinguish normal input from abnormal input

SQL Injection

Countermeasures – Defensive Coding

- Parameterized query insertion prevents SQLi by allowing developers to:
  - more accurately specify the structure of a SQL query
  - pass value parameters to it separately
    - any unsanitary user input is not allowed to modify the query structure
- SQL DOM
  - Is a set of classes that enables automated data type validation and escaping
    - To escape is to take the data we already have and secure it prior to rendering it for the end user
    - Escaping converts the special HTML characters to HTML entities so that they are displayed, instead of being executed.
  - Encapsulates database queries to provide a safe and reliable way to access databases
  - This changes the query-building process from
    - an unregulated one that uses string concatenation to a systematic one that uses a type-checked API
  - Within the API, developers are able to systematically apply coding best practices such as input filtering and rigorous type checking of user input

SQL Injection

Countermeasures – Detection Methods

- Signature based
  - Attempts to match specific attack patterns
  - Such an approach must be constantly updated and may not work against self-modifying attacks
- Anomaly based
  - Attempts to define normal behavior and then detect behavior patterns outside the normal range
  - In general terms, there is a training phase, in which the system learns the range of normal behavior, followed by the actual detection phase
- Code analysis
  - Code analysis techniques involve the use of a test suite to detect SQLi vulnerabilities
  - The test suite is designed to generate a wide range of SQLi attacks and assess the response of the system

SQL Injection

innovate achieve lead

SQLi Attack Reports

- Imperva Web Application Attack Report – July 2013
  - Surveyed a cross-section of Web application servers in industry and monitored eight different types of common attacks
  - The report found that SQLi attacks ranked first or second in total number of attack incidents, the number of attack requests per attack incident, and average number of days per month that an application experienced at least one attack incident
  - Imperva observed a single Web site that received 94,057 SQL injection attack requests in one day
- The Open Web Application Security Project Report – 2013
  - On the ten most critical Web application security risks listed injection attacks, especially SQLi attacks, as the top risk
  - This ranking is unchanged from its 2010 report.
- The Veracode State of Software Security Report – 2013
  - found that percentage of applications affected by SQLi attacks is around 32% and that SQLi attacks account for 26% of all reported breaches
  - Veracode also considers this among the most dangerous threats, reporting that three of the biggest SQL injection attacks in 2012 resulted in millions of email addresses, user names, and passwords being exposed and damaged the respective brands.
- The Trustwave Global Security Report – 2013
  - lists SQLi attacks as one of the top two intrusion techniques
  - The report notes that poor coding practices have allowed the SQL injection attack vector to remain on the threat landscape for more than 15 years, but that proper programming and security measures can prevent these attacks

Source: Information Security Principles & Practice by William Stallings & Lawrie Brown

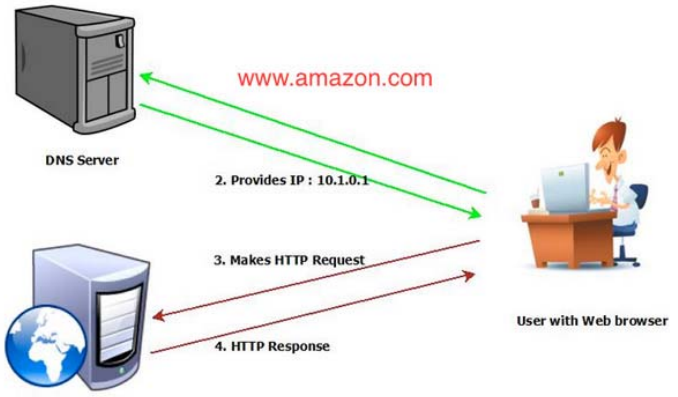
DNS Tunneling Attack

Domain Name System

innovate achieve lead

Overview

- The Domain Name System (DNS) is a huge phone book or White Pages for the Internet
- We, humans, access information online through domain names
  - E.g., www.nytimes.com or www.yahoo.com
- DNS translates domain names to IP addresses so browsers can load Internet resources
- Each device connected to the Internet has a unique IP address which other machines use to find the device
- We type the domain name (E.g., www.amazon.com) in our browser
- The browser then queries the DNS server for amazon.com's IP address
- The DNS returns Amazon's IP address
- The browser then requests data from Amazon's web host using the IP Address



Source: https://www.cloudflare.com/learning/dns/what-is-dns/

Domain Name System

innovate achieve lead

Domain Name System Terminology

- Domain Names
  - A domain name is a human-readable name—like amazon.com
  - The Internet Corporation for Assigned Names and Numbers (ICANN) manages these domain names
- Top Level Domain (TLD)
  - TLD refers to the last part of a domain name
    - For example, the .com in amazon.com is the Top Level Domain
  - Most common TLDs include .com, .net, org, and .info
  - Country code TLDs represent specific geographic locations
    - For example: .in represents India
- Second Level Domain
  - This is the part of a domain name which comes right before the TLD—amazon.com—for example.
- Sub Domain
  - A subdomain can be created to identify unique content areas of a web site. For example, the aws of aws.amazon.com.

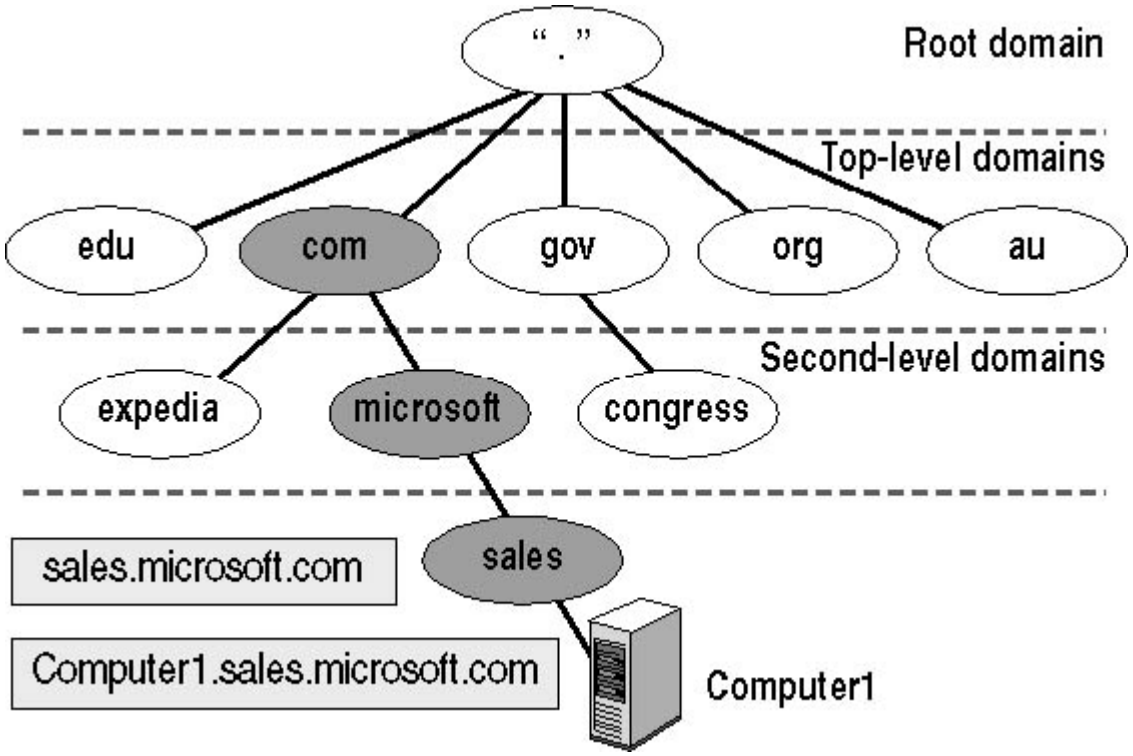
- Some examples of TLDs:
  - com – Commercial businesses.
  - gov – U.S. government agencies.
  - edu – Educational institutions such as universities.
  - org – Organizations (mostly non-profit).
  - mil – Military.
  - net – Network organizations.
  - in - India
  - eu – European Union.

Source: https://cloudacademy.com/blog/how-dns-works/

Domain Name System

innovate achieve lead

DNS Hierarchy

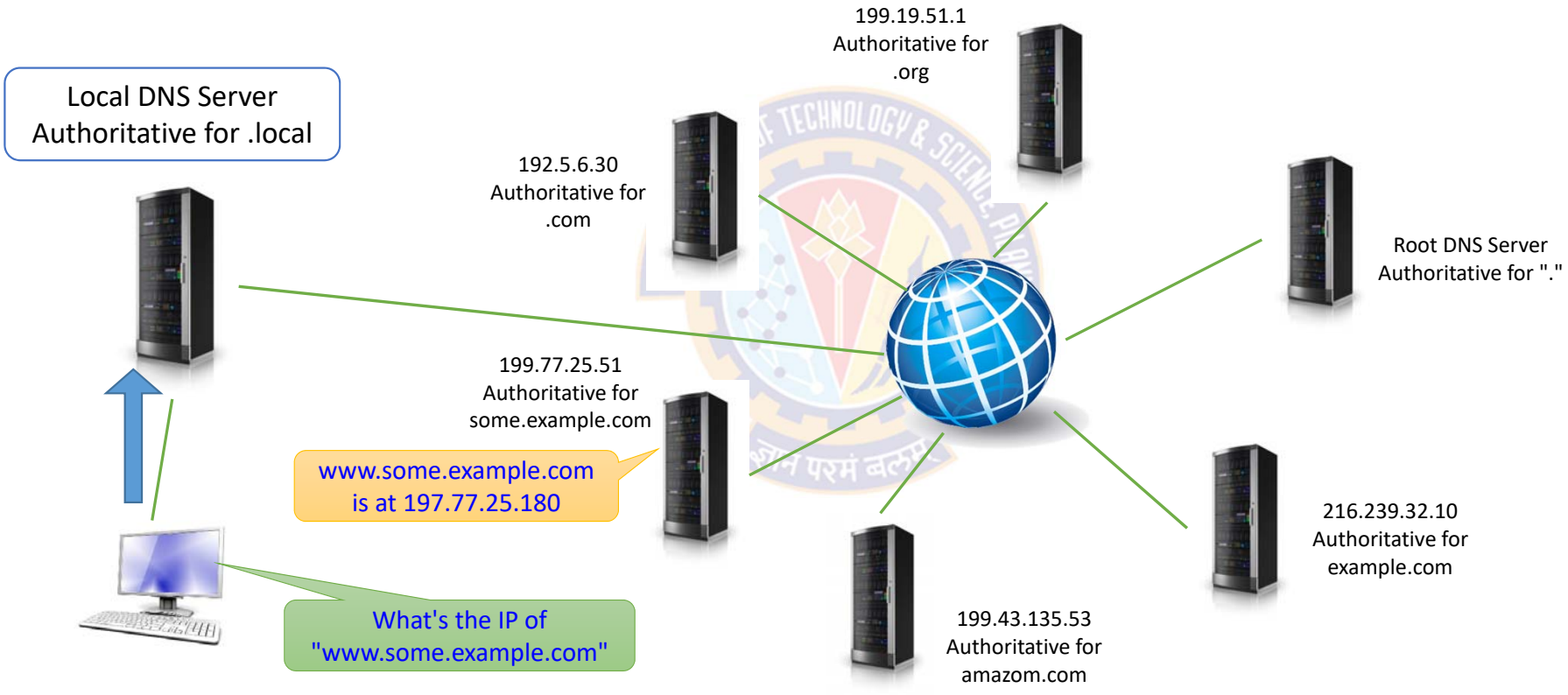


Source: http://etutorials.org/

Domain Name System

innovate achieve lead

Resolving Domain Names



Source: YouTube: Bypassing Firewalls with DNS Tunnelling

DNS Tunneling Attack

Overview

- DNS Tunneling turns DNS or Domain Name System into a hacking weapon
- DNS supports a fairly simple query-response protocol
  - This protocol allows admins to query a DNS server's database
- Hackers secretly communicate with a target computer by sneaking in commands and data into the DNS protocol
  - This idea is at the core of DNS Tunneling

Source: https://www.varonis.com/blog/dns-tunneling/

DNS Tunneling Attack

What is DNS Tunneling

- Attack Objective
  - Use the DNS protocol itself as a data transport mechanism, taking advantage of DNS's free flow through firewalls
- Attack Impact
  - Covert channel for data exfiltration or otherwise avoiding "standard" communication paths
- Attack forms
  - Software within one domain encodes data into DNS resource records for transmission through firewalls
  - DNS "query" sent to tunnel endpoint DNS server
  - Tunnel endpoint DNS server "answers" with response data

Source: https://www.varonis.com/blog/dns-tunneling/

DNS Tunneling Attack

innovate achieve lead

What is DNS Tunneling

- Suppose hackers were in control of the DNS server
- Hackers can fake responses and send data back to the target system
- They can return messages hidden in various DNS response fields to the malware they loaded on the victim's computer
  - For example, directing the malware to search this folder, etc.
- The "tunneling" part of this attack is about obscuring the data and commands to avoid detection by firewall (monitoring software)
- Hackers can use base32, base64 or other character sets, or even encrypt the data
- This encoding would get past simple detection software that's searching on plaintext patterns

Source: https://www.varonis.com/blog/dns-tunneling/

DNS Tunneling Attack

innovate achieve lead

Resolving Domain Names

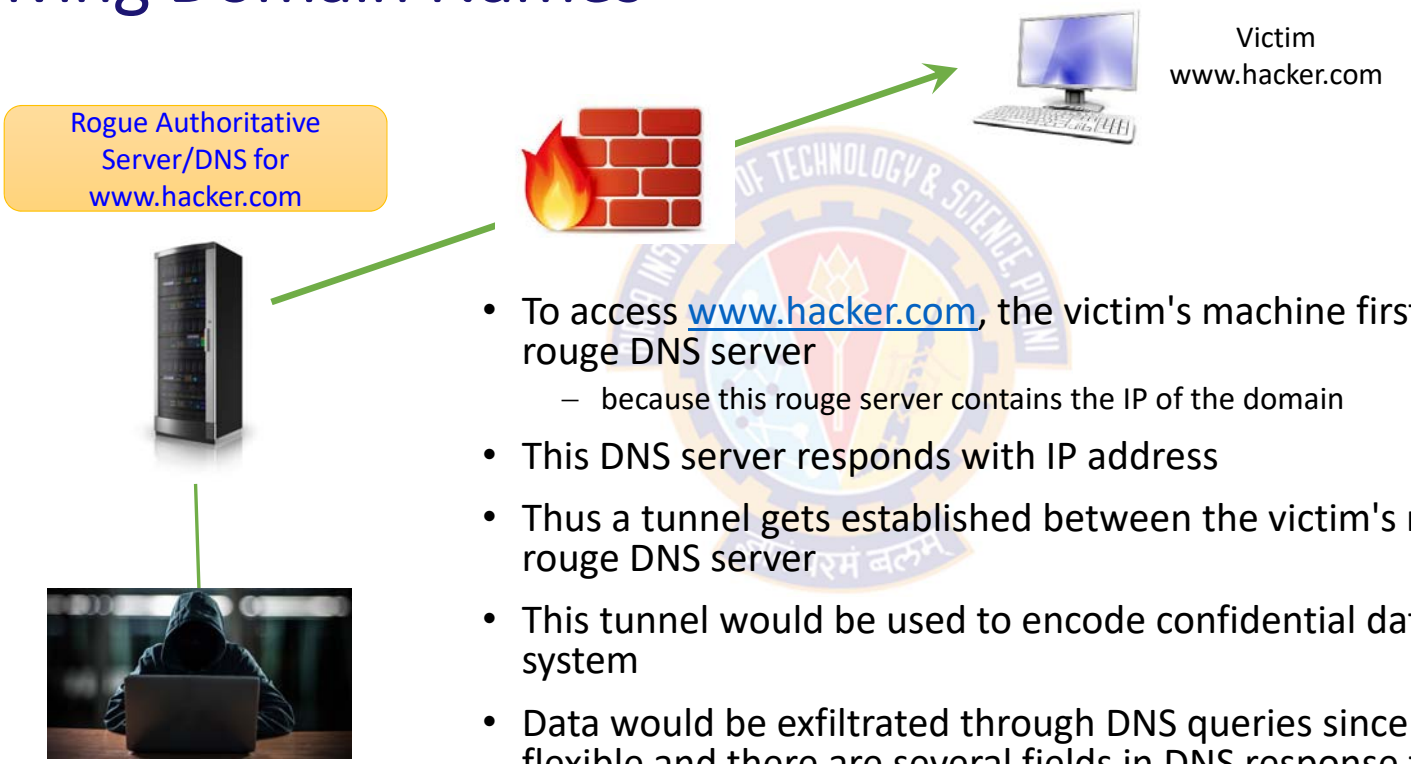
- The hacker sets up a rouge authoritative/DNS server
- This server contains mapping of IP addresses of websites set up by the hacker
- The hacker installs a malware in the victim's computer
- This malware tries to find the IP address of the website setup by the hacker (E.g., www.hacker.com)
- To get the IP address, the malware uses DNS protocol, which would be passed through the firewall

Source: YouTube: Bypassing Firewalls with DNS Tunnelling

DNS Tunneling Attack

innovate achieve lead

Resolving Domain Names



- To access www.hacker.com, the victim's machine first approaches the rouge DNS server
  - because this rouge server contains the IP of the domain
- This DNS server responds with IP address
- Thus a tunnel gets established between the victim's machine and the rouge DNS server
- This tunnel would be used to encode confidential data of the internal system
- Data would be exfiltrated through DNS queries since this protocol is very flexible and there are several fields in DNS response to exfiltrate data

Source: YouTube: Bypassing Firewalls with DNS Tunnelling

DNS Tunneling Attack

innovate achieve lead

What is DNS Tunneling

- DNS tunneling allows:
  - Data Exfiltration
    - Hackers sneak sensitive data out over DNS
    - It's certainly not the most efficient way to get data from a victim's computer— with all the extra overhead and encoding—but it can work and it's stealthy!
  - Command and Control (C2)
    - Hackers can also use the DNS protocol to send simple commands to, say, a Remote Access Trojan (RAT)
  - IP-Over-DNS Tunneling
    - There are utilities that have implemented the IP stack on the DNS query-response protocol
    - This would make it relatively easy to transfer data using standard communications software like FTP, Netcat, ssh, etc.,.

Source: https://www.varonis.com/blog/dns-tunneling/

DNS Tunneling Attack

innovate achieve lead

DNS Tunneling Utilities

- If you want to do your own pen testing to see how well your company can detect and respond, there are a few utilities available
- All the ones below do IP-over-DNS:
  - Iodine
    - Available on many platforms (Linux, Mac OS, FreeBSD, and Windows)
    - Lets you set up an SSH shell between the target and the route computer
  - OzymanDNS
    - Dan Kaminsky's DNS tunneling project written in Perl
    - You can SSH with it.
  - DNSCat2
    - "A DNS tunnel that won't make you sick"
    - Creates an encrypted C2 channel to let you upload/download files, run a shell, etc.

Source: https://www.varonis.com/blog/dns-tunneling/

DNS Tunneling Attack

Detecting DNS Tunneling

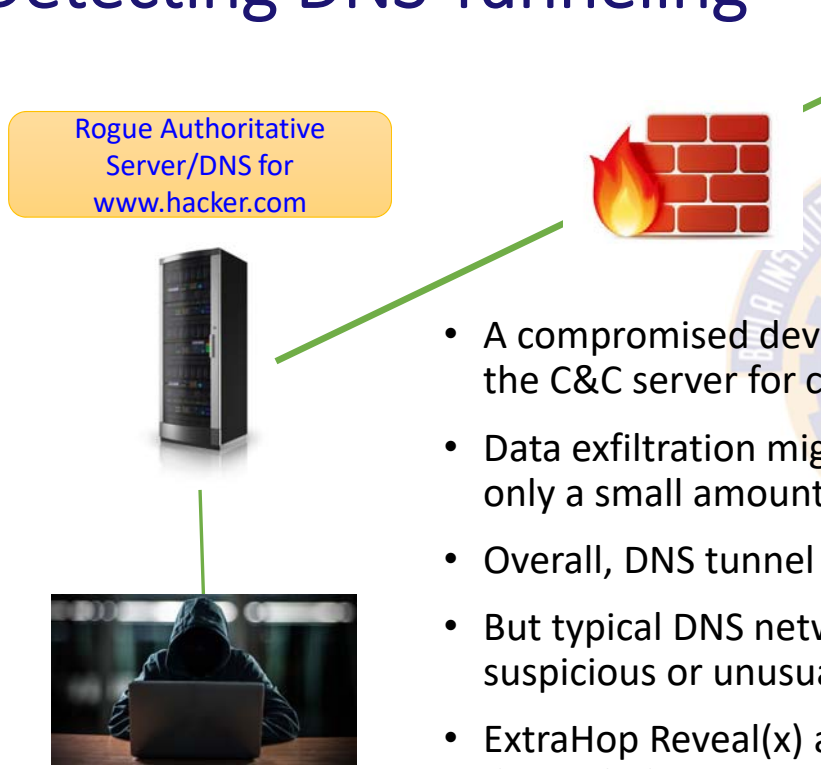
- Two ways to detect DNS Misuse
  - Payload Analysis
    - Involves looking for unusual data being sent back and forth using statistical techniques. For example:
      - strange-looking hostnames
      - a DNS record type that's not used all that often
      - unusual character sets that can be spotted by statistical techniques
  - Traffic Analysis
    - Involves looking at the number of DNS domain requests and comparing it against the normal usage baseline
    - Hackers who are performing DNS tunneling will create very heavy traffic to the server
      - In theory, much greater than a normal DNS exchange. And that should be detectable!

Source: https://www.varonis.com/blog/dns-tunneling/

DNS Tunneling Attack

innovate achieve lead

Detecting DNS Tunneling



- A compromised device might continuously send DNS queries to beacon, or ping, the C&C server for commands
- Data exfiltration might also result in many DNS queries because a query can hold only a small amount of data
- Overall, DNS tunnel traffic can be noisy
- But typical DNS network traffic can also be noisy, making it difficult to distinguish suspicious or unusual DNS queries from the legitimate ones
- ExtraHop Reveal(x) automatically detects unusual changes in DNS traffic based on device behavior over time, surfacing queries that should be investigated

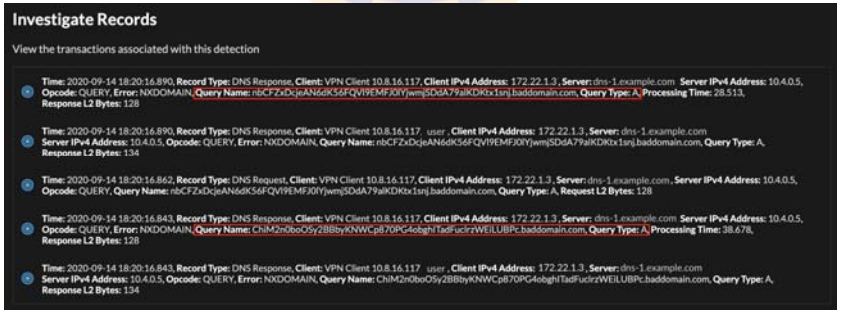
Source: YouTube: Bypassing Firewalls with DNS Tunnelling

DNS Tunneling Attack

innovate achieve lead

Detecting DNS Tunneling

- Investigate DNS Queries
  - DNS queries include record types, which help DNS servers retrieve the correct information about the requested domain
  - For example, an "A" record type requests the IPv4 address for a domain name
  - Attackers might encode data into a subdomain name of an A record type, such as base64encodedgibberish.baddomain.com:

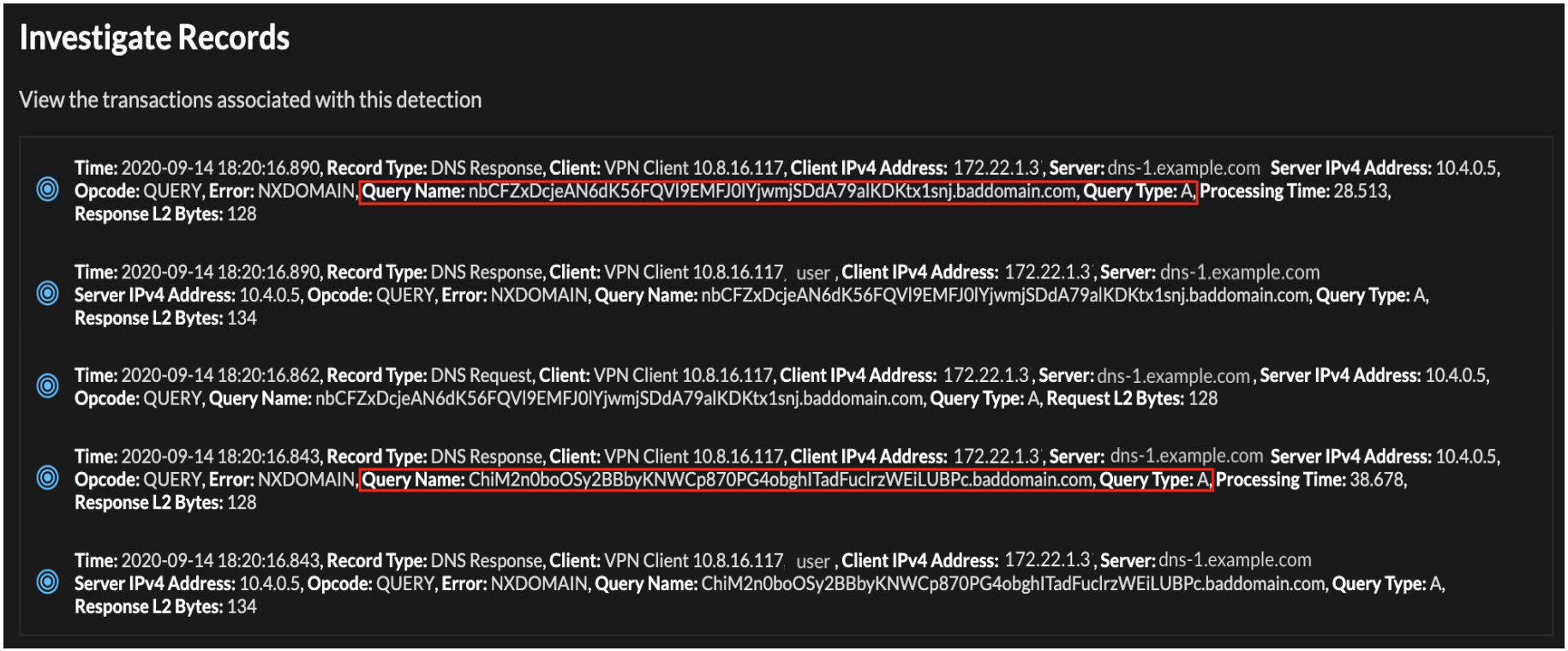


Source: https://www.extrahop.com/company/blog/2020/dns-tunneling-definition-and-protection/

DNS Tunneling Attack

innovate achieve lead

Detecting DNS Tunneling



Source: https://www.extrahop.com/company/blog/2020/dns-tunneling-definition-and-protection/

DNS Tunneling Attack

innovate achieve lead

Detecting DNS Tunneling

- How to Prevent DNS Tunneling
  - Because DNS is an essential service, it can be difficult to block
  - But a defender might be able to identify suspicious domains and IP addresses from threat intelligence matches
  - DNS traffic that is sent to known malicious endpoints and domains can be blocked at the perimeter
  - Also, internal clients can be configured to send all queries to an internal DNS server that filters or blocks suspicious domains
  - Keep in mind that suspicious domains might be short-lived to avoid detection
  - Staying vigilant for suspicious domains, and reporting suspicious domains to threat intelligence platforms, can help reduce the effectiveness of DNS tunnels in abetting malicious C&C activity

Source: https://www.extrahop.com/company/blog/2020/dns-tunneling-definition-and-protection/

DNS Tunneling Attack

innovate achieve lead

Detecting DNS Tunneling

- ExtraHop Reveal(x) automatically detects unusual changes in DNS traffic based on device behavior over time, surfacing queries that should be investigated
- A defender can investigate the DNS query from the detection card
- Investigate DNS Queries
- DNS queries include record types, which help DNS servers retrieve the correct information about the requested domain. For example, an "A" record type requests the IPv4 address for a domain name. Attackers might encode data into a subdomain name of an A record type, such as base64encodedgibberish.baddomain.com:

Source: https://www.extrahop.com/company/blog/2020/dns-tunneling-definition-and-protection/


innovate
achieve
lead

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
ज्ञानं परमं बलम्

Thank You!