Topic 3.2 - Authentication and Web Security

Eric Casanovas

Universitat d'Andorra

16th March, 2022

Introduction

- Refers to the detect, prevent and respond cybersecurity threads related to web
- Websites can be seen as "secure" but far from we think, they have to be protected
- Many website attacks can be performed, but first it's important to see some securizations to websites

What is DNSSec? I

- DNSSec comes from Domain Name System Security Extensions
- According to Wikipedia: Is a suite of extension specifications by the Internet Engineering Task Force (IETF) for securing data exchanged in the Domain Name System (DNS) in Internet Protocol (IP) networks
- The proper functioning of the Internet is critically dependent on the DNS
- *Remember: DNS translates URLs to IPs*
- It does not provide privacy, but prevents manipulations or poisoned responses from DNS.

What is DNSSec? II

- With DNSSec we avoid:
  - DNS poisoning
  - DNS cache poisoning
  - DNS Hijacking

DNS Recap I

OSI model

Layer	Name	Example protocols
7	Application Layer	HTTP, FTP, DNS, SNMP, Telnet
6	Presentation Layer	SSL, TLS
5	Session Layer	NetBIOS, PPTP
4	Transport Layer	TCP, UDP
3	Network Layer	IP, ARP, ICMP, IPSec
2	Data Link Layer	PPP, ATM, Ethernet
1	Physical Layer	Ethernet, USB, Bluetooth, IEEE802.11

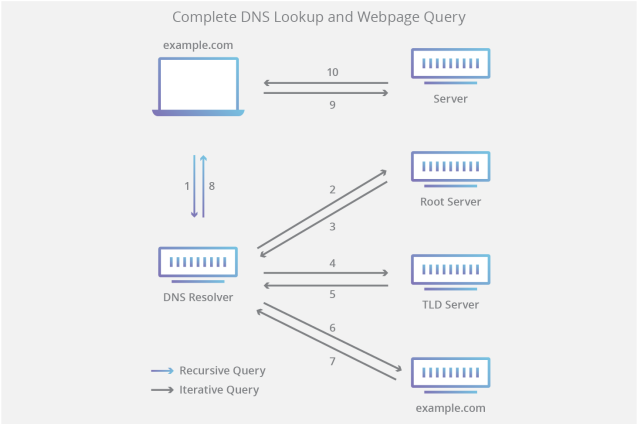
DNS Recap II

Normal DNSs

To ask for "example.com":

1. Computer asks DNS resolver to resolve "example.com"
2. DNS Resolver asks the root server the TLD server
3. Root server answers with the IP of TLD server
4. DNS Resolver asks TLD server (.com server) for example.com
5. TLD server answers with exapmle.com ip
6. DNS Resolver asks example.com
7. example.com answers with the ip of the resource
8. DNS resolver answers to Computer
9. Computer can communicate with example.com resources

DNS Recap III

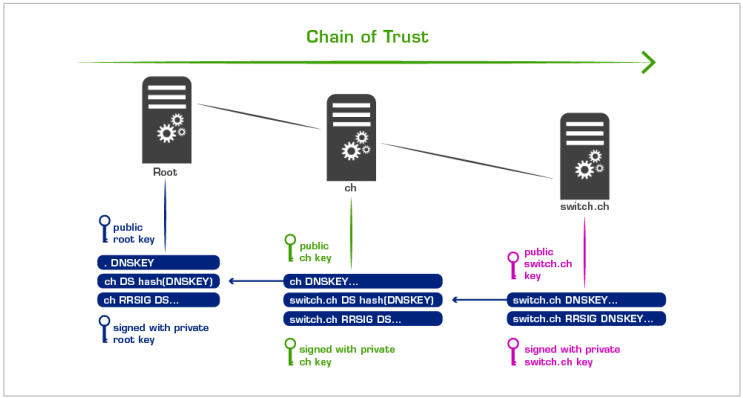


How DNSSec works? I

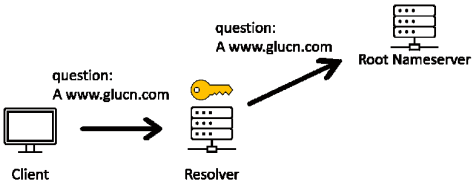
1. RRSIG: Contains a cryptographic signature
2. DNSKEY: Contains Public signing key
3. DS: Contains the hash of a DNSKEY record

How DNSSec works? II

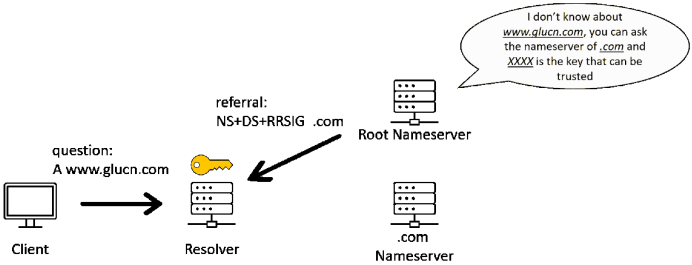
Chain of trust:



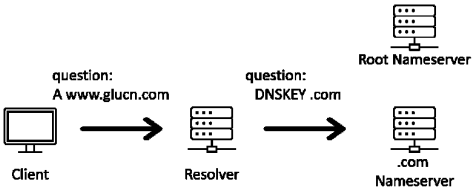
How DNSSec works? III



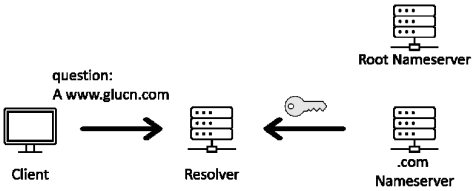
How DNSSec works? IV



How DNSSec works? V

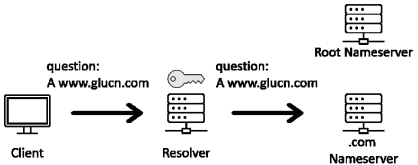


How DNSSec works? VI

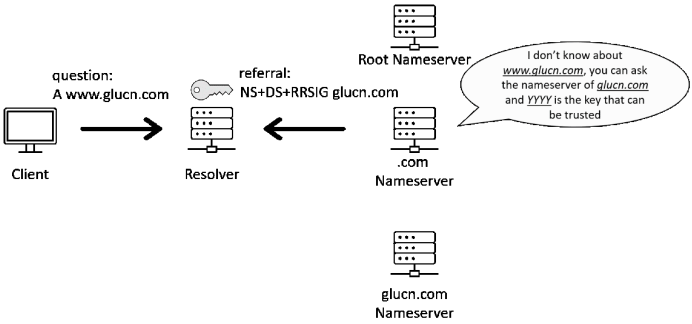


How DNSSec works? VII

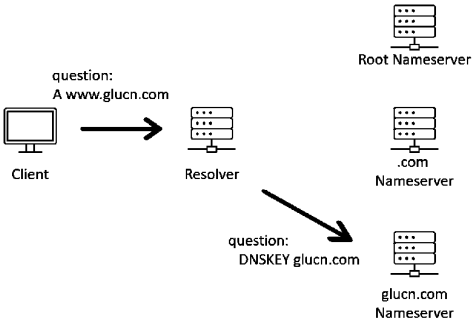
- Check DNSKEY with DS given by root nameserver



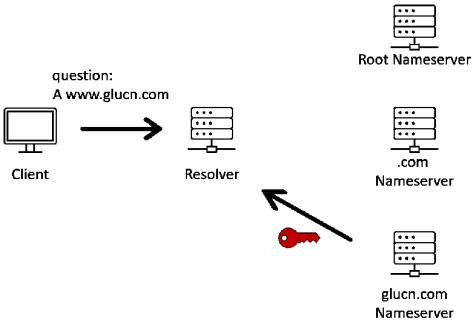
How DNSSec works? VIII



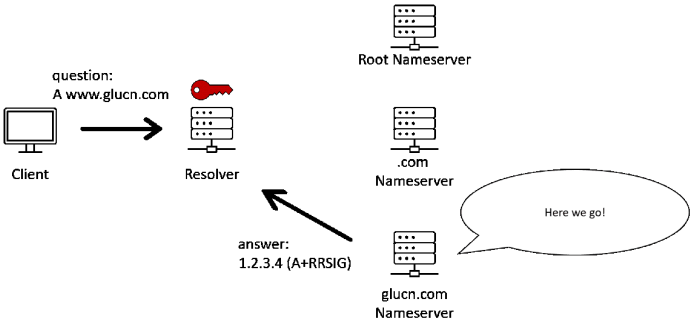
How DNSSec works? IX



How DNSSec works? X

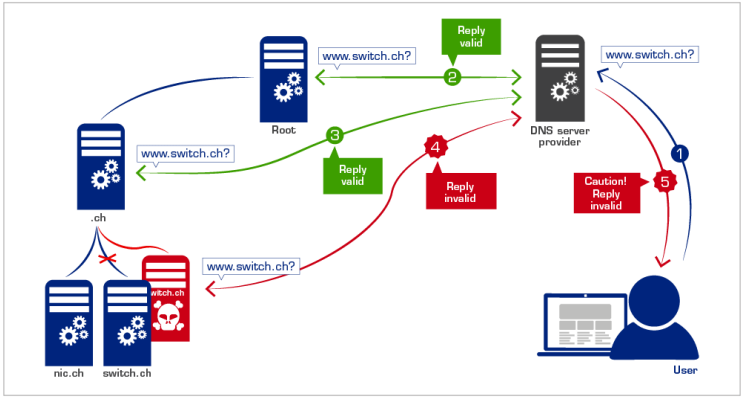


How DNSSec works? XI



How DNSSec works? XII

In case of malicious actor:



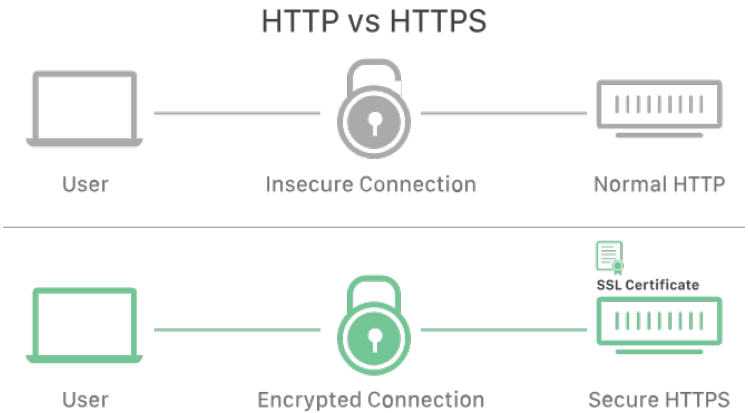
DNSSec strengths and weaknesses

- DNSSEC protects against MITM attacks
- DNSSEC protects against DNS Poisoning
- DNSSEC does not protect the privacy of DNS
- DNSSEC can't protect against DNS local caches!

SSL & TLS I

- SSL stands for Secure Socket Layer
- TLS stands for Transport Layer Security
- TLS is the successor of SSL
- They are cryptographic protocols to secure communications in computer networks
- SSL developped in 90s and released 1995
- SSL evolved to TLS in 1999
- TLS 1.0 1999 (deprecated), TLS 1.1 2006 (deprecated), TLS 1.2 2008, TLS 1.3 2018

SSL & TLS II


HTTP vs HTTPS
User
Insecure Connection
Normal HTTP
SSL Certificate
User
Encrypted Connection
Secure HTTPS

SSL & TLS III

OSI model

Layer	Name	Example protocols
7	Application Layer	HTTP, FTP, DNS, SNMP, Telnet
6	Presentation Layer	SSL, TLS
5	Session Layer	NetBIOS, PPTP
4	Transport Layer	TCP, UDP
3	Network Layer	IP, ARP, ICMP, IPSec
2	Data Link Layer	PPP, ATM, Ethernet
1	Physical Layer	Ethernet, USB, Bluetooth, IEEE802.11

How TLS Works? I

- Using Asymmetric Cryptography to:
  - Encrypt/Decrypt secret keys (protect ephemeral secret key agreements)
  - Create digital signatures
  - Certificates!
- Using Symmetric Cyptography to:
  - Encrypt/Decrypt messages
  - Authenticate packets

How TLS Works? II

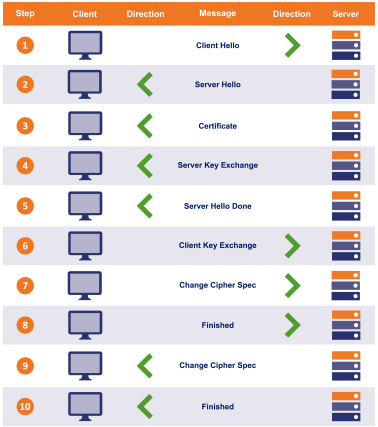
- TLS can be divided in 2 steps:
  - TLS Handshake
  - Data Exchange

How TLS Works? III

TLS1.2:

1. Client->Server: Connection Request (TCP)
2. Server->Client: Connection Acknowledged (TCP)
3. Client->Server: Client Hello (TLS)
4. Server->Client: ServerHello + Certificate (TLS)
5. Client->Server: ClientKeyExchange + CipherSpec + Finished (TLS)
6. Server->Client: CipherSpec + Finished (TLS)
7. Client->Server: Asks for data (Data Exchange)
8. Server->Client: Answers (Data Exchange)

How TLS Works? IV



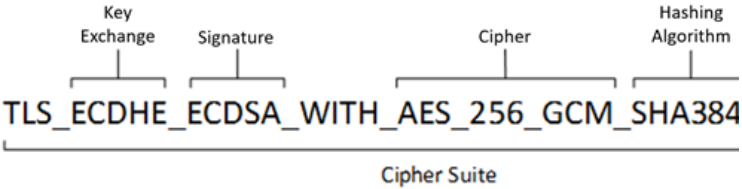
How TLS Works? V

TLS1.3:

1. Client->Server: Connection Request (TCP)
2. Server->Client: Connection Acknowledged (TCP)
3. Client->Server: Client Hello + List Supported Cipher Suites + Guess Key Agreement Protocol and Key Share (TLS)
4. Server->Client: ServerHello + Key Agreement Protocol + Key Share + Finished (TLS)
5. Client->Server: Check Certificate + Generate Key + Finish (TLS)
6. Client->Server: Asks for data (Data Exchange)
7. Server->Client: Answers (Data Exchange)

How TLS Works? VI

How TLS Works? VII



Introduction I

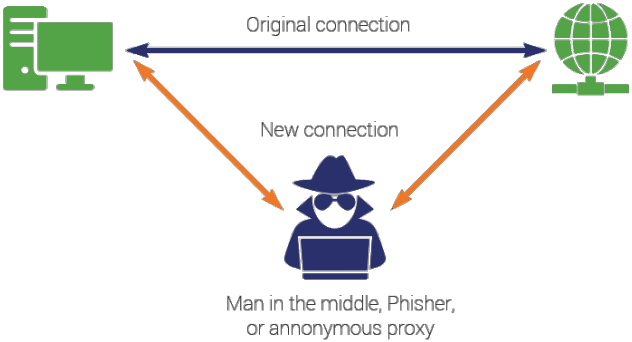
- Even with securizations websites are vulnerable to attacks
- The threats range from human errors to sophisticated attacks
- The goal for attacker:
  - Gain unauthorized access
  - Obtain information
  - Introduce malicious content
  - Alter website

Introduction II

Example:

- MITM
- Cross-Site Scripting (XSS)
- Injection attacks
- DoS/DDoS
- Bruteforce
- Zero-day
- File upload attacks

MITM I


Original connection
New connection
Man in the middle, Phisher,
or annonymous proxy

MITM II

- Trying to put in the middle of a communication
- In web the attacker will try to get the shared key
- To perform this attack the attacker has to generate certificates for client and server
- To avoid this kind of attack -> Signed certificates!
- Browsers also helps notifying when certificates are not signed by a trusted CAs

MITM III

Your connection is not private

Attackers might be trying to steal your information from **self-signed.badssl.com** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

Help improve Safe Browsing by sending some system information and page content to Google.
Privacy policy

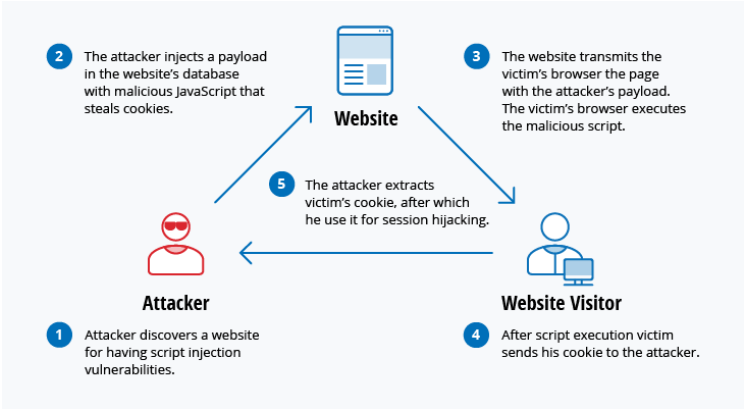
Advanced

Back to safety

XSS I

- XSS attacks use third-party web resources to run scripts in the victim's web browser
- The attacker injects a payload with malicious javascript
- When victim asks for the website, it contains the malicious script that is executed in the victim's machine
- Often used to steal cookies, but can be used to exploit additional vulnerabilities

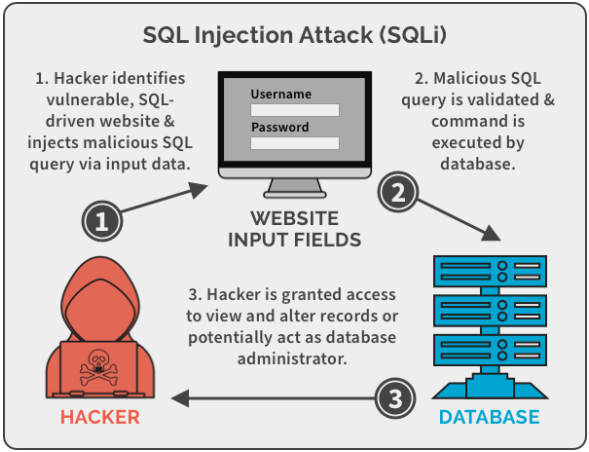
XSS II


2 The attacker injects a payload in the website's database with malicious JavaScript that steals cookies.
Website
3 The website transmits the victim's browser the page with the attacker's payload. The victim's browser executes the malicious script.
5 The attacker extracts victim's cookie, after which he use it for session hijacking.
Attacker
Website Visitor
1 Attacker discovers a website for having script injection vulnerabilities.
4 After script execution victim sends his cookie to the attacker.

Injection Attacks I

- Consists in injecting code to get data
- The most common is SQLInjection
- Focused mainly in retrieve "hidden" data from databases thanks to vulnerabilities in the code
- Very important to code properly database controllers
- NEVER do something like:
  SELECT * FROM table WHERE user = $USER_INPUT
  Easy attack introducing: 1 OR 1 == 1

Injection Attacks II



DoS & DDoS I

- Consists in overwhelm system resources so that it cannot respond to service requests
- A DDoS is launched from a large number of host machines
- Many types of DoS attacks:

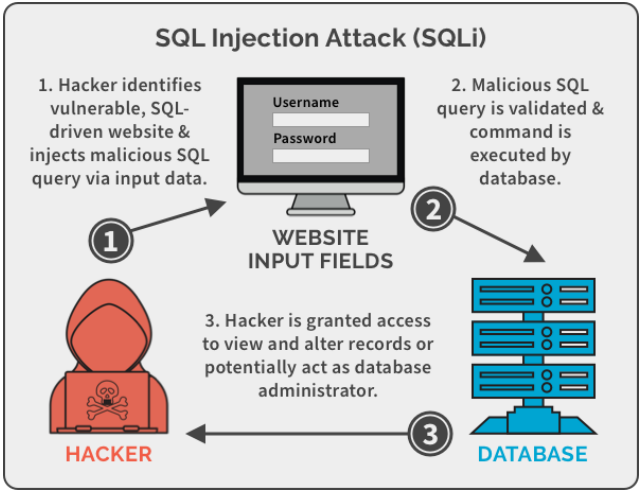
DoS & DDoS II

- TCP SYN flood attack:
  - Send many Sync TCP packets to flood the servers
  - Solution: Place servers behind a firewall configured to stop inbound Sync packets.
- Teardrop attack:
  - The attacker sends fragmented packets to the target server, and in some cases where there's a TCP/IP vulnerability, the server is unable to reassemble the packet, causing overload and crashing.
  - Solution: Often focused on ports 139 and 445 (SMB), so closing this ports it would be enough

DoS & DDoS III

- Ping of death attack:
  - Consists in sending pings to the target machine larger than 65535 bytes. IP packets larger than 65535 are not allowed, so they are fragmented and one reassembled can cause buffer overflows and crashes
  - Solution: Block large ping packets
- Botnets:
  - Consists in control many machines to perform an attack to overwhelm the target host
  - Solution: Filtering traffic

DoS & DDoS IV


SQL Injection Attack (SQLi)
1. Hacker identifies vulnerable, SQL-driven website & injects malicious SQL query via input data.
Username
Password
2. Malicious SQL query is validated & command is executed by database.
1
2
WEBSITE INPUT FIELDS
3. Hacker is granted access to view and alter records or potentially act as database administrator.
3
HACKER
DATABASE

Bruteforce

- Consists in make random guesses until get a password, user login, etc...
- This kind of attacks are performed to login checkboxes in webpages or offline if the attacker has a hash of a password or key.
- To avoid this kind of attacks is very important to use the appropriate encryption/hashing algorithms

File Upload Attacks

- Consists in exploiting the ability to post information to a website to upload malicious files
- This attack is very dangerous as the attacker can introduce any file to the victim's server

Zero-Day Attacks

- Consists in attacks that are not known yet
- This kind of attack is the most dangerous one because the attacker knows the vulnerability but the victim does not
- The way to protect from zero-day is to update software immediately after a new release

The END!