Network security for developers

CYDNet3d | 3 days | On-site or online | Hands-on

In our connected world, networked applications are more exposed to cyberattacks than ever - therefore, securing the communication between the system's components is extremely important.

This course focuses on the "whys" and "hows" of secure communication. It provides foundational knowledge about essential cryptographic algorithms and their usage (hashing, encryption, digital signatures, PKI), and puts them into practice in a TCP/IP environment through practical exercises. Starting from the use of secure sockets and TLS certificate management, you'll see a systematic overview of network attacks on each layer of the OSI model from data link to application. This includes classic attacks against IPv4 and IPv6 networks like ARP and NDP spoofing, DHCP starvation and SYN floods as well as more modern application-layer attacks such as DNS cache poisoning and Slowloris.

Of course the course also covers appropriate best practices and recommendations to prevent these attacks, from secure switch configuration to secure operating system settings and the proper use of secure protocols on each layer.

Because even if you don't know about these attacks, the hackers certainly will!

Skills and drills

20 LABS

6 CASE STUDIES

Audience

Developers working on networked applications

Group size

12 participants

Preparedness

General network application development, networking basics

Outline

- Cyber security basics
- Cryptography for developers
- Network security
- Denial of service
- Security by design
- Wrap up

Standards and references

OSI model

What you'll have learned

- Getting familiar with essential cyber security concepts
- Understanding how cryptography supports security
- Understanding the most common attacks from OSI Layer 2 to Layer 7
- Using network traffic manipulation tools

Table of contents

Day 1

› Cyber security basics

What is security?
Threat and risk
Cyber security threat types – the CIA triad
Cyber security threat types – the STRIDE model
Consequences of insecure software
Constraints and the market

› Cryptography for developers

Cryptography basics

Elementary algorithms

- Hashing
  - Hashing basics
  - *Case study - Shattered*
  - Common hashing mistakes
  - *Lab – Hashing*
- Hash algorithms for password storage
  - Password storage algorithms and considerations
  - Best practices when using password hashing algorithms
- Random number generation
  - Pseudo random number generators (PRNGs)
  - Cryptographically secure PRNGs
  - Seeding
  - Using virtual random streams
  - *Lab – Using random numbers*
  - True random number generators (TRNG)
  - Assessing PRNG strength
  - *Case study – Equifax credit account freeze*

Confidentiality protection

- Symmetric encryption

- Stream ciphers
- Block ciphers
- Modes of operation
- Modes of operation and IV – best practices
- Best practices
- Best practices – Using cryptographic storage

Lab – Symmetric encryption

- Asymmetric encryption
  - The RSA algorithm
    - Using RSA – best practices

    Case study – RSA attacks: Bleichenbacher, ROBOT, and Marvin
- Combining symmetric and asymmetric algorithms
- Key exchange and agreement
  - Key exchange
  - Diffie-Hellman key agreement algorithm
  - Key exchange pitfalls and best practices

Integrity protection

- Authenticity and non-repudiation
- Message Authentication Code (MAC)

Lab – Calculating MAC

Day 2

› Cryptography for developers (continued)

Integrity protection (continued)

- Digital signature
  - Digital signature with RSA
  - Elliptic Curve Cryptography
    - ECC basics
    - ECC curves and pitfalls
    - Digital signature with ECC

    Lab – Digital signature with ECDSA
  - Authenticated encryption
    - Authenticated encryption modes of operation
    - Authenticated encryption modes of operation: CCM
    - Authenticated encryption modes of operation: GCM

Public Key Infrastructure (PKI)

- Key management challenges

- Certificates
  - Certificates and PKI
  - Chain of trust
  - X.509 certificates
  - PKI actors and procedures
  - Enrollment and identification
  - Inappropriate certificate validation
  - PGP – Web of Trust
  - Certificate pinning
  - Certificate revocation
  - *Lab – Certificate chain validation vulnerabilities*
  - *Lab – CA certificate pinning*

› Network security

Network security overview

The communication layers

Threats against TCP/IP

The Data Link layer

- ARP spoofing and ARP poisoning
- *Lab – ARP spoofing*
- Protecting against ARP spoofing
- Attacks against the Spanning Tree Protocol
- Mitigating Spanning Tree Protocol attacks
- MAC flooding and MAC table overflow
- Port stealing
- Port protection
- Data link attacks and IPv6
- *Lab – IPv6 NDP spoofing*
- DHCP threats
- *Lab – DHCP starvation*
- Protecting against DHCP attacks
- DHCPv6 security
- VLAN issues
- Securing VLANs
- Sniffing
- Protecting your network against sniffing and MitM

The network layer

- Spoofing IP addresses
- Protecting against IP spoofing
- IP fragmentation and the teardrop attack
- IPv6-specific attacks and defenses
- Smurf attack against ICMP

🔬 *Lab – Smurf attack*

📖 *Case study – Ping of death*

- Redirecting ICMP – route hijacking

🔬 *Lab – Route hijacking*

- Black hole attacks and selective forwarding in ad hoc networks
- Attacks against ICMPv6
- Routing protocol threats
- Securing routing protocols
- IPsec overview
- IPsec usage scenarios and typical mistakes
- IPsec cryptographic requirements

Day 3

› Network security (continued)

The transport layer

- The TCP protocol
- The UDP protocol
- SYN flooding

🔬 *Lab – SYN flooding*

- Protecting against SYN floods
- UDP flooding
- TCP session hijacking and other attacks
- Fingerprinting via TCP, UDP, and ICMP

🔬 *Lab – Fingerprinting and service detection*

- Firewalls and IDS

🔬 *Lab – Using a NIDS*

The application layer

- The Domain Name System

- DNS cache poisoning

Lab – DNS cache poisoning

- DNS rebinding
- DNS amplification
- DoS targeting DNS
- Securing DNS systems

Lab – DNSSEC

Case study – MaginotDNS attack

- Secure protocols
- Securing email protocols
- Web application firewalls and IDS

Transport security

- The TLS protocol
  - TLS basics
  - TLS features (changes in v1.3)
  - The handshake in a nutshell (v1.3)
  - An example TLS 1.3 handshake
  - TLS best practices

  Lab – Using a secure socket
  - HTTP Strict Transport Security (HSTS)
- Securing HTTP
  - From HTTP to HTTPS
  - MitM proxies
  - Dangerous HTTP methods and headers
  - Slow DoS attacks against HTTP

  Lab – Slowloris

› Denial of service

Flooding

Resource exhaustion

Sustained client engagement

Infinite loop

Case study – DoS against Tesla GUI via malicious web page

Economic Denial of Sustainability (EDoS)

Algorithmic complexity issues

- Regular expression denial of service (ReDoS)

  Lab – ReDoS

- Dealing with ReDoS

› Security by design

Secure design principles of Saltzer and Schroeder

- Economy of mechanism
- Fail-safe defaults
- Complete mediation
- Open design
- Separation of privilege
- Least privilege
- Least common mechanism
- Psychological acceptability
- Additional principles
  - Work factor
  - Compromise recording

› Wrap up

Secure coding principles

- Principles of robust programming by Matt Bishop

And now what?

- Software security sources and further reading
- Network security resources