

Computer Network and Security Principles (CNSP)

Prof. Dr. Burkhard Stiller¹, Prof. Dr. Bruno Rodrigues², Dr. Alberto Huertas¹, Thomas Grübl¹

¹Communication Systems Group CSG
Department of Informatics Ifi, University of Zürich UZH
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

²University of St. Gallen, Embedded Sensing Systems ICV-HSG, Dornbirn, Austria



Universität
Zürich

Phone: +41 44 635 6710
E-Mail: [stiller|huertas|gruebl@ifi.uzh.ch]
bruno.rodrigues@unisg.ch



Assistants:

Thomas Grübl, Anderson Rocha, Dr. Alberto Huertas
Phone: +41 44 635 [4337|7585|7585], FAX: +41 44 635 6809
E-Mail: [gruebl|rocha|huertas@ifi.uzh.ch]

Lecture Objectives

- Understand** basic principles of network security (such as confidentiality, integrity, availability), major threats, cryptography mechanisms (symmetric and asymmetric), essentials of cloud security, existing security guidelines and risk assessment frameworks, as well as to understand the intersection between Machine Learning (ML) and Cybersecurity
- Identify** basic and advanced (i.e., basics of penetration testing) network security problems and propose associated solutions, being necessary the evaluation and classification of risk, as well as the association of basic mitigation measures
- Apply** basic security concepts (such as security policies, access control mechanisms, firewalls and intrusion detection systems) in different network architectures and protocols, as well as detailing advanced security mechanisms for communications

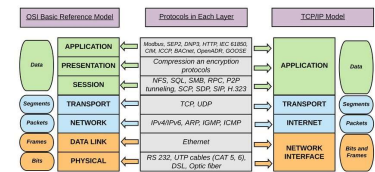
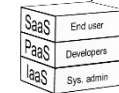
Competencies

- During this class students are expected to enhance their **competencies** to
 - Understand** and **identify** major threats toward the design, development, and prototyping of security policies;
 - Enhance situational awareness
 - Manage** security in computer networks;
 - Learn to monitor network activities actively (traffic monitoring, packet inspection) or passively (e.g., logs, forensics)
 - Develop the ability to use (and design) specific security tools
 - Deploy** mitigation approaches in case of certain attacks
 - Be able to enforce preventive or reactive measures against threats identified

Prerequisites ... Ideally

- Basics of networking architectures and protocols
 - OSI reference model and TCP/IP model
 - Cloud computing
 - Security principles
 - Confidentiality
 - Integrity
 - Availability

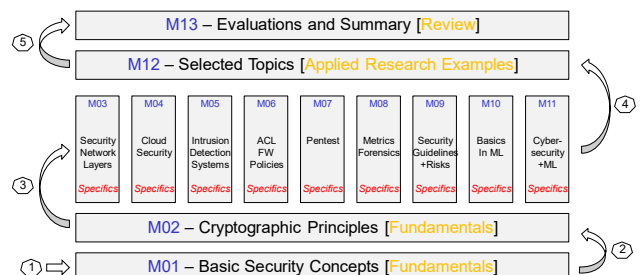
Cloud computing



Lecture Outline (Plan)

Lecture	Date	Lecturer	Content
M01	19.09	BS	Course Organization, Basic Security Concepts
M02	26.09	BS	Cryptographic Principles
M03	03.10	BS	Security on Different Network Layers
M04	10.10	BR	Cloud Security
M05	17.10	BR	Intrusion Detection and Prevention
M06	24.10	BR	Access Control, Firewalls, Policies
M07	31.10	BS+External	Penetration Tests
M08	07.11	BR	Security Metrics and Network Forensics
M09	14.11	TG	Network Security Guidelines and Risk Management
M10	21.11	AH	Basics of Machine Learning
M11	28.11	AH	Cybersecurity and Machine Learning
M12	05.12	External	Selected Cybersecurity Aspects
M13	12.12	BR	Evaluations and Summary
CT	19.12	Students	Challenge Task Presentations

Overview of Lecture (Chapter Dependency)



Teaching Aids and Challenge Task

- Lecture slides
 - <https://www.csg.uzh.ch/csg/en/teaching/HS24/CNSP.html>
 - Slides accessible ONLY with a UZH-internal IP address (VPN)
- Reading material
 - Text Books
 - William Stallings: Network Security Essentials: Applications and Standards (6th Edition 2016), Pearson, 978-0-134-52733-8
 - William Stallings: Cryptography and Network Security: Principles and Practice (8th Edition 2022), Pearson, 978-1-292-43748-4
 - To read corresponding book chapters
 - Lecture slides only provide a basic overview on different topics



– Exercises and Challenge Task (CT)

Lecture's Background (References)

- Michael Steward: Network Security, Firewalls, and VPNs, ISBN 10: 076379130X
- Peter Mell, Timothy Grance. The NIST Definition of Cloud Computing. Special Publication 800-145. National Institute of Standards and Technology (NIST). 2011.
- Patrick Engebretson. The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy. Elsevier, 2013.
- Xin, Yang, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng Gao, Haixia Hou, and Chunhua Wang. "Machine Learning and Deep Learning Methods for Cybersecurity." IEEE Access 6 (2018): 35365-35381.
- Khan, Suleman, Abdullah Gani, Ainuddin Wahid Abdul Wahab, Muhammad Shiraz, and Iftikhar Ahmad. "Network Forensics: Review, Taxonomy, and Open Challenges." Journal of Network and Computer Applications 66 (2016): 214-235.

Computer Network and Security Principles (CNSP)

M1 – Basic Security Concepts

Basic Security Concepts

- Objectives
 - To provide a general understanding on the course contents and structure
 - To overview basics of networks and distributed systems
 - To understand the interconnection between essential topics of security in networks
- Topics
 - Course overview and structure
 - Network Basics
 - Security Pillars
 - Security Service and Mechanisms
 - Security Design Principles
 - Trust in Cyber-physical Systems

NETWORK BASICS

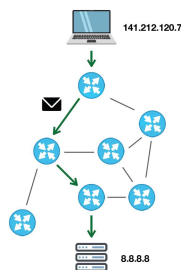
Definition and Types

- A definition of “*Network*”
 - Group of **devices** (e.g., computers, smartphones, tablets) **connected by a transmission** media to share data and provide services based on their available resources
- Types of networks
 - Personal Area Network (**PAN**)
 - Local Area Network (**LAN**)
 - Metropolitan Area Network (**MAN**)
 - Wide Area Network (**WAN**)
 - Others: Global/Interplanetary Area Network (**GAN, IAN**)
 - **Internet** (irrestricted connections) and **Intranet** (restricted connections)



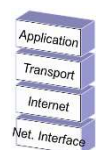
The Internet

- Global network that provides **best-effort** delivery of packets between connected hosts
 - **Packet**: structured sequence of bytes
 - **Header**: metadata used by the network
 - **Payload**: user data to be transported
- Every host has a **unique identifier**, the IP address
- Series of routers receive packets, look at destination address on the header and send it one op towards the destination IP address



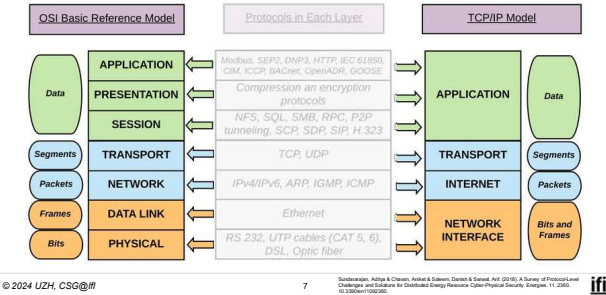
Communication Protocols

- Hosts communicate based on clearly defined network protocols
 - **Syntax**: How communication is structured (e.g., format and order of messages)
 - **Semantics**: What communication means. Actions taken on transmit or receipt of message, or when a timer expires. What assumptions can be made
- Networks use a stack of **protocol layers**
 - Each layer has different responsibilities
 - Layers define abstraction boundaries:
 - Lower layers provide services to layers above
 - Higher layers use services of layers below



Protocols Stack

- Two predominant views
 - OSI (theoretical) and TCP/IP (practical)



© 2024 UZH, CSG@#

7

Boudier, Adria & Chen, Armin & Salami, Davood & Senechal, Arif (2016). A Survey of Protocol-Level Challenges and Solutions for Distributed Energy Resource Cyber-Physical Security. *Designs* 10, 2000. <https://doi.org/10.3390/design10102000>

ifi

Sample Attacks within OSI Layers

OSI Layers	Functions	Example Protocol	Example of Attacks
Application	User interface, process to application	HTTP, FTP, IMAP, SMTP	Malware, DoS, SMTP Attacks, Buffer Overflows
Presentation	Data format, encryption	SSL, TLS, JPEG, MPEG	Phishing, session sniffing
Session	Process-to-process communication	SQL, RPC, NFS	Session Hijacking, Telnet and FTP sniffing
Transport	End-to-end communication maintenance	TCP, UDP	Desynchronization, TCP sequence prediction, SYN flooding, Denial-of-Service (DoS)
Network	Routing data, logical addresses, WAN	IP, ICMP	Sybil attacks, IP spoofing, port sniffing, Distributed Denial of Service (DDoS), Man-in-the-middle (MITM)
Data Link	Physical addressing, LAN	ARP	MAC Spoofing, MAC/ARP sniffing
Physical	Transmission of bytes	802.11, Ethernet, Bluetooth	Sniffing, Jamming signals, tampering data

© 2024 UZH, CSG@#

8

ifi

Malicious Code

- I.e., **Malware**: One of the biggest security threats, which is broadly defined as

A program that is covertly inserted into another program with the intent to destroy data, run destructive or intrusive programs, or otherwise compromise the confidentiality, integrity, or availability of the victim's data, applications, or operating system. NIST SP 800-61

- There are different **variations** of malware
 - Viruses, worms, trojans, ransomware, spyware, adware, phishing
- A malicious code can be either active or passive
 - Active**: involves the non-authorized altering of the host system, compromising or rendering the system unusable
 - Passive**: targets social engineering, conduct, and social aspects
 - Some authors use a third-class named **grayware**
 - Code that is not necessarily malicious but monitors or scans collecting personal information

© 2024 UZH, CSG@#

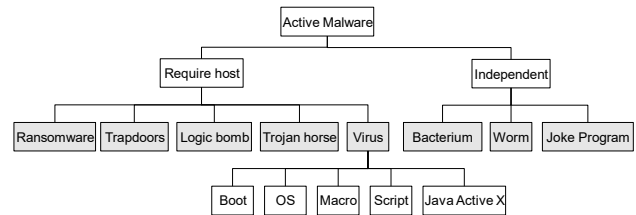
9

<https://github.com/ByteHackr/MalwareAnalysis>

ifi

Active Malware

- Typically alter a target system, affecting all major security pillars: confidentiality, integrity, and availability



© 2024 UZH, CSG@#

10

ifi

Examples of Active Malware

- Virus**: program that include itself in one or more files and execute an action
- Trojan horse**: program with an explicit functionality (known by the user) and an unknown functionality (unknown by the user)
- Logic bomb**: program that execute a malicious action upon an external event
 - Empty the payment roll if an employee X is fired
- Trapdoor (backdoor)**: vulnerability intentionally coded for posterior system access
- Ransomware**: program that restricts (by encrypting files) access to another system (e.g., storage). Access is blocked until a ransom is paid by the victim
- Worm**: program that replicates itself from one system or host to another, typically via network
- Joke program**: non-damaging, non-spreading, and mostly harmless programs that are not intended to cause damage to computers.
- Bacter**: program that exhausts resources of the host system (CPU, RAM, Disk)
 - `while (1) { fork(); }`

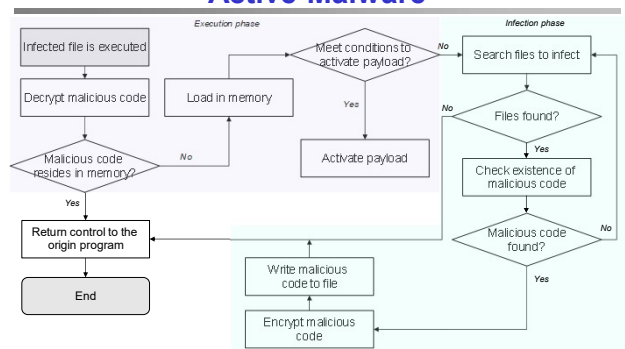
© 2024 UZH, CSG@#

11

http://web.deu.edu.tr/docoreilly/networking/pubs/ch11_01.htm

ifi

Sample Functioning of a Typical Active Malware



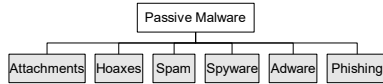
© 2024 UZH, CSG@#

12

ifi

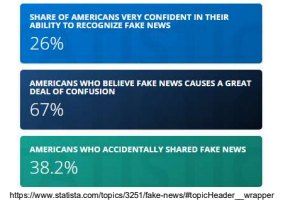
Passive Malware

- Its construction may or not involve the development of a new, independent malicious code; it can be based on existing code. Necessarily involve social engineering aspects



Examples of Passive Malware

- Hoax:** spread of fake message, i.e., fake news
- Spam:** non solicited messages aiming to sell a service or product
- Spyware:** programs that monitor and store the user behavior
 - Keylogger
- Adware:** programs that shows advertisements
- Phishing:** use of fraudulent methods to retrieve sensitive information, such as passwords
 - Spear phishing:** phishing targeting specific individuals



A Not-So-Common Cold: Malware Statistics in 2022

<https://dataprot.net/statistics/malware-statistics/>

SECURITY PILLARS

Security Basics (1)

- Computer security is defined by NIST as follows

Computer Security: The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications)

- The NIST definition introduces three key objectives, known as the **CIA triad**

- Confidentiality** based on two types of confidentiality



Data confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized individuals

Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed

Security Basics (2)

- Integrity** based on two types of integrity

Data integrity: Assures that information (both stored and in transmitted packets) and programs are changed only in a specified and authorized manner

System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system

- Availability**

Assures that systems work promptly, and service is not denied to authorized users



Accountability and Authenticity

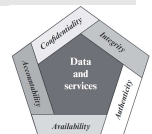
- Although the CIA triad is well established, some in the security field feel that additional concepts are needed

- **Accountability**

The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action. Because truly secure systems are not yet an achievable goal, we must be able to trace a security breach to a responsible party. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes

- **Authenticity**

The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source



Additional Pillars

- Stallings define **five** major security pillars for Computer Networks
 - Confidentiality, Integrity, Availability, Accountability, and Authenticity
- However, other authors mention additional ones
 - **Authorization**
 - Partner with a known ID is enabled to use a service
 - **Non-repudiation**
 - Neither the sender or the receiver can deny that a communication happened
 - **Anti-replay protection**
 - Protects a received from duplicated reception of a previously obtained and authenticated message

Examples (1)

- Which security concept is violated?
 - User A transmits a file to user B. The file contains sensitive information (e.g., payroll records) that is to be protected from disclosure. **User C**, who is not authorized to read the file, **is able to monitor the transmission and capture a copy of the file** during its transmission.
 - Confidentiality
 - A network manager D transmits a message to a computer E, under its management. The message instructs computer E to update an authorization file to include the identities of a number of new users who are to be given access to that computer. **User F intercepts the message, alter its contents to add or delete entries**, and then forwards the message to computer E, which accepts the message as coming from manager D and updates its authorization file accordingly.
 - Integrity

Examples (2)

- Rather than intercept a message, **user F constructs its own message with the desired entries and transmits that message to computer E as if it had come from manager D**. Computer E accepts the message as coming from manager D and updates its authorization file accordingly.
 - Authenticity
- An employee is fired without warning. The personnel manager sends a message to a server system to invalidate the employee's account. When the invalidation is accomplished, the server is to post a notice to the employee's file as confirmation of the action. **The employee is able to intercept the message and delay it long enough to make a final access to the server to retrieve sensitive information**. The message is then forwarded, the action taken, and the confirmation posted. The employee's action may go unnoticed for some considerable time.
 - Availability
- A message is sent from a customer to a stockbroker with instructions for various transactions. Subsequently, the investments lose value and **the customer denies sending the message**.
 - Accountability

Defining “Threat” and “Attack”

- In the literature, the terms *threat* and *attack* are commonly used to mean more or less the same thing
 - RFC 4949, *Internet Security Glossary*, provides the following definitions

Threat: A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

Attack: An assault/action on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

- Threats precede attacks and attacks are exploiting threats

Defining “Risk”, “Threat”, and “Vulnerability”

- An attack is an exploitation of a threat, but what is the difference between **risk**, **threat**, and **vulnerability**?

Threat: A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability. (RFC 4949, *Internet Security Glossary*)

Risk: A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence. (NIST SP 800-30 Rev. 1)

Vulnerability: Weakness in a system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat. (NIST SP 1800-17b)

- Vulnerability → System weakness
- Threat → Potential malicious actions exploiting vulnerabilities
- Risk → Likelihood of threats to be exploited
 - Threat X Vulnerabilities OR Likelihood X Impact
- Attack → Exploitation of a threat

IT Security defines a process of risk management, supported by a set of suitable technical measures!

Defining “Security” and “Safety”

- Security (refresh)

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications)

- Safety:

Freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment. NIST SP 800-160 Vol.2 Rev-1

SECURITY MECHANISMS AND SERVICES

Security Attacks

- A useful means of classifying security attacks, used both in ITU-T X.800 and RFC 4949, is in terms of passive attacks and active attacks
 - Passive attacks:** attempts to learn or make use of information from the system but does not affect system resources
 - Release/disclosure of confidential message contents**
 - Traffic analysis:** observing communication patterns
 - Active attacks:** attempts to alter system resources or affect their operation
 - Masquerade:** entity pretending to be a different entity
 - Replay:** passive capture and subsequent retransmission
 - Modification of messages:** alter, delay, or reorder messages
 - Denial-of-service:** disruption of a service

Security Services (2)

Security Service	Description	Specific Services
Authentication	The assurance that the communicating entity is the one that it claims to be	Peer Entity Authentication: used in association with a logical connection to provide confidence in the identity of the entities connected Data-origin Authentication: in a connectionless transfer, provides assurance that the source of received data is as claimed
Access control	The prevention of unauthorized use of a resource	
Data confidentiality	The protection of data from unauthorized disclosure	Connection Confidentiality: the protection of all user data on a connection Connectionless Confidentiality: the protection of all user data in a single data block Selective-field Confidentiality: the confidentiality of selected fields within the user data on a connection or in a single data block Traffic-flow Confidentiality: the protection of the information that might be derived from observation of traffic flows
Data integrity	The assurance that data received are exactly as sent by an authorized entity	Connection Integrity with Recovery: provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted Connection Integrity Without Recovery: as above, but provides only detection without recovery Selective-Field Connection Integrity: Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed Connectionless Integrity: Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided Selective-Field Connectionless Integrity: Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified
Non-repudiation	Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication	Non-repudiation Origin: Proof that the message was sent by the specified party Non-repudiation Destination: Proof that the message was received by the specified party

OSI Security Architecture

- The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined (*ITU-T3 Recommendation X.800*) briefly as
 - Security attack:** Any action that compromises the security of information owned by an organization
 - Security mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack
 - Security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service

Security Services (1)

- Two definitions of security service are mentioned in Stallings¹

A security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers. (ITU-T X.800)

A processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms. (RFC 4949, Internet Security Glossary)

- X.800 divides these services into five categories and fourteen specific services:
 - Authentication, Access Control, Data Confidentiality, Data Integrity, and Non-repudiation

Security Mechanisms (1)

- Mechanisms are divided into those implemented in a **specific** protocol layer, such as TCP or an application-layer protocol, and those that are not specific to any particular protocol layer or security service (**pervasive**)
- A security service may use one or more security mechanisms

SERVICE	MECHANISM					
	Peer entity authentication	Data origin authentication	Confidentiality	Traffic flow confidentiality	Data integrity	Nonrepudiation
Peer entity authentication	Y	Y		Y		
Data origin authentication	Y	Y				
Access control			Y			
Confidentiality			Y			
Traffic flow confidentiality			Y	Y		
Data integrity			Y	Y	Y	
Nonrepudiation			Y	Y		Y
Availability				Y	Y	

Security Mechanisms (2)

Specific Security Mechanism: May be incorporated into the appropriate protocol layer to provide some OSI security service	Pervasive Security Mechanism: Mechanisms that are not specific to any particular OSI security service or protocol layer
Encipherment: The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.	Trusted Functionality: That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).
Digital Signature: Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).	Security Label: The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.
Access Control: A variety of mechanisms that enforce access rights to resources.	Event Detection: Detection of security-relevant events.
Data Integrity: A variety of mechanisms used to assure the integrity of a data unit or stream of data units	Security Audit Trail: Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
Authentication Exchange: A mechanism intended to ensure the identity of an entity by means of information exchange.	Security Recovery: Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.
Traffic Padding: The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.	
Routing Control: Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.	
Notarization: The use of a trusted third party to assure certain properties of a data exchange.	

SECURITY DESIGN PRINCIPLES

Fundamental Security Design Principles

- In the absence of such foolproof security services and mechanisms, it is useful to have a set of **widely agreed design principles** that can guide the development of protection mechanisms¹:
1. Economy of mechanism
 2. Fail-safe defaults
 3. Complete mediation
 4. Open design
 5. Separation of privilege
 6. Least privilege
 7. Least common mechanism
 8. Psychological acceptability
 9. Isolation
 10. Encapsulation
 11. Modularity
 12. Layering
 13. Least astonishment

The first eight listed principles were first proposed in Saltzer and Schroeder (1975) and have withstood the test of time.

Security Design Principles (1)

1. **Economy of mechanism**: the design should be relatively as simple and small as possible
 - Rationale: easier to test and verify thoroughly in contrast to complex and large mechanisms
2. **Fail-safe defaults**: access decisions should be based on permission rather than exclusion
 - Rationale: a mechanism that gives explicit permission tends to fail by refusing permission, a safe situation that can be quickly detected.
3. **Complete mediation**: every access must be checked against the access control mechanism
 - Rationale: systems should not rely on caches to authorize users

Security Design Principles (2)

4. **Open design**: the design of a security mechanism should be open rather than secret
 - Rationale: Algorithms can be reviewed by many experts, and users can, therefore, have higher confidence in them
5. **Separation of privilege**: multiple privilege attributes are required to achieve access to a restricted resource (e.g., multi-factor authentication)
 - Rationale: useful to mitigate the potential damage of a computer security attack
6. **Least privilege**: every process and every user of the system should operate using the least set of privileges necessary to perform the task
 - Rationale: reduce impact of potential unwanted actions

Security Design Principles (3)

7. **Least common mechanism**: minimize the functions shared by different users, providing mutual security.
 - Rationale: reduce the number of unintended communication paths and reduces the amount of hardware and software on which all users depend
8. **Psychological acceptability**: security mechanisms should not interfere unduly with the work of users, while at the same time meeting the needs of those who authorize access
 - Rationale: security mechanisms should be transparent to the users of the system or at most introduce minimal obstruction

Security Design Principles (4)

Design Principle	Description	Rationale
9. Isolation	Public resources should be isolated from critical resources to prevent disclosure or tampering	Separation of concerns.
10. Encapsulation	Specific form of isolation based on object-oriented functionality. Isolates objects within a domain.	Encapsulates a collection of procedures and data objects within a domain.
11. Modularity	The development of security functions as separate, protected modules and to the use of a modular architecture for mechanism design and implementation.	Provide common security functions and services, such as cryptographic functions, as common modules.
12. Layering	Refers to the use of multiple, overlapping protection approaches addressing the people, technology and operational aspects of information systems	By using a layered approach, the failure or circumvention of any individual protection approach will not leave the system unprotected.
13. Least astonishment	A program or user interface should always respond in the way that is least likely to astonish the user.	Avoid unexpected application behavior.

TRUST IN CYBER-PHYSICAL SYSTEMS

Toward a “Trust” Definition

□ Trust is...

“...a **confidence in the integrity** of an entity for reliance on that entity to fulfil specific responsibilities...” ETSI GS NFV-SEC

“... a particular level of the **subjective probability** with which an agent will perform a particular action...” Gambetta, Diego (2000)

“...**firm belief** in the reliability, truth, or ability of someone or something.” Lexico (Dictionary)

“...as the **belief** that the other party will **behave in a socially responsible manner**, and, by so doing, will fulfill the trusting party’s expectations without taking advantage of its vulnerabilities.” Pavlou, Paul A. (2003)

- It can be summarized as a **perception or expectation of consistency over time** for a particular task

$$\text{Trust} = \frac{\text{Consistency}}{\text{Time Period}}$$



Trust in Cyber Physical Systems

- Often analyzed in terms of **CIA** (Confidentiality, Integrity, and Availability)

– **Trust** is (often) a **perception** of CIA’s parameters of a particular system



- A cloud-based service often requires data Confidentiality, Integrity and Availability
- Trust in a CSP can be defined on the perception of a customer of CIA parameters

- However, it is not straightforward to quantify **CIA**, or security in general
- Heavily dependent on customer perception

Trust Evaluation

- There are available methods for evaluation trust whereas the choice relies on **available resources, threats, and risks**

– Look-up tables

- Tracking behavior based on an expected outcome

– Reputational

- Evaluation across a set of elements leading to a calculation of reputation

– Game theoretical

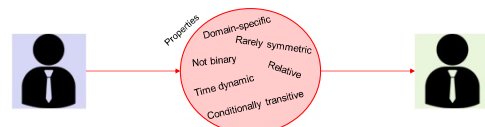
- Mapping of relations as a game trying to achieve a Nash equilibrium

– Probabilistic

- Based on a variable behavior

Trust Models

- Refers to the type of trusting relationship that can exist between individuals and entities



- Trust relationships can happen in two ways

– Based on a **direct trust**

- Direct **relation** between entity **A** and entity **B**

– Based on a **trusted intermediary**, i.e., **trusted third-party**

- Not always direct **relation** between **A** and **B** based on a **centralized entity** or **decentralized system**

Trust based on an Intermediary

- Trust based on a **trusted intermediary**
 - Entity A initially do not trust entity B, but trust another entity or system C
 - Centralized:** Trusted Authority (TA) that can establish the basis for the negotiation between two entities



- Distributed:** A Public Permissionless Blockchain where entities transact based on a transparent and verifiable intermediary



Trust in a Decentralized System

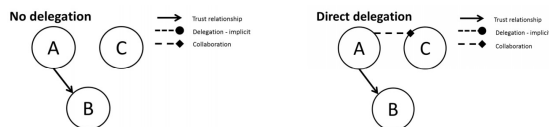
- Entities requires a **uniquely identifiable identity**
 - Trust is subject to a consistency in a time period, a reputable identity takes effort to be built

$$\text{Trust} = \frac{\text{Consistency}}{\text{Time Period}}$$

- Ideally, all actions should be **transparent** and **verifiable** (logged)
 - When these conditions are met, it is possible to reduce the time period to create a trustable relationship
- Each entity should have its **own policies** to make take decisions for itself
 - Trust is **highly subjective**, general rules to create trust are less likely to be accepted

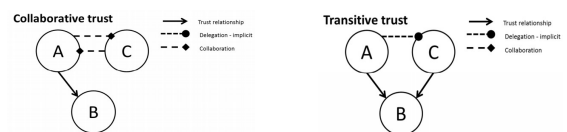
Propagation of Trust in a Decentralized System (1)

- When the trusting entity delegates to an entity the decision of whether it should trust an entity between them in the chain
- Delegation** (conditionally transitive trust)
 - Entity A is unable to evaluate the appropriate level of trust for a relationship with another entity B



Propagation of Trust in a Decentralized System (2)

- Collaborative** involves two more entities (entities A and C) working together to decide whether to trust another (entity B)



- Transitive** is the decision by an entity A to trust entity B because entity C trusts it

Trust or not to Trust

- Trust is **highly subjective**, but measurable in some cases
 - Security, in terms of major pillars, is not possible to quantified
- Blockchain is one of the **greatest enabler of trust in a decentralized systems**
 - Entities are not necessarily needed to know each other, as long as, all actions (transactions) are publicly verifiable
 - Enablers of zero trust systems
- However, the use of **Trusted-third Party (TTP)** is useful in most existing cases of centralized and distributed systems

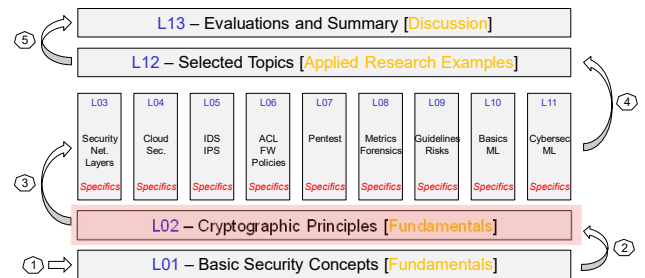
Major Takeaways

- Comprehension of network security principles require a definition of **"network"** and **"security"**
 - OSI provides a theoretical view on the network stack
 - Security basics in terms of CIA (Confidentiality, Integrity, Availability)
- Security services and mechanisms**
 - Association of which mechanisms are used in which services based on different design principles
- Trust** plays a major role in security being often seen as a subjective **perception** of security services and mechanisms

Computer Network and Security Principles (CNSP)

M02 – Cryptographic Principles

Overview of Class and Lectures (Chapter Dependency)



Cryptographic Principles

- Objectives
 - To provide a general overview of encryption mechanisms, approaches, and use cases
 - To understand basics of symmetric encryption
 - To understand basics of asymmetric encryption
 - To overview certificates and digital signatures
- Topics
 - Basics overview of encryption and decryption
 - Symmetric encryption model and techniques (rotation, transposition, rotor machines, steganography)
 - Asymmetric encryption (crypto model, RSA, elliptic curve, applications)
 - Certificates and Digital Signature

ENCRYPTION MECHANISMS

Basic Definitions (1)

- An original message is known as the **plaintext**, while the coded message is called the **ciphertext** 
- The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**
 - Restoring the plaintext from the ciphertext is **deciphering** or **decryption**.
- The many schemes used for encryption constitute the area of study known as **cryptography**
 - Such a scheme is known as a **cryptographic system** or a **cipher** 
- Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis** 

Basic Definitions (2)

- An encryption scheme is said to be **computationally secure** if
 - The **cost** of breaking the cipher **exceeds the value of the encrypted information** 
 - The **time** required to break the cipher **exceeds the useful lifetime of the information** 
- An encryption scheme is **unconditionally secure** if
 - The ciphertext does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available
 - **Regardless of the time**, an attacker cannot decrypt the ciphertext because required information is not available

Encryption and Decryption

- En-/Decoding (En-/Decryption), Ciphering (En-/Decipher) of data to ensure confidentiality and privacy

- **Encryption** of plain text

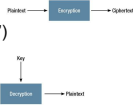
- Only possible with the knowledge of a key (the “secret”)
- Easy to do and fast to process

- **Decryption** of cipher text (encrypted data)

- Only successful with the right key
- Extremely large, dedicated, and specific calculation effort, iff the key is not known (attack situation only), otherwise easy and fast to process

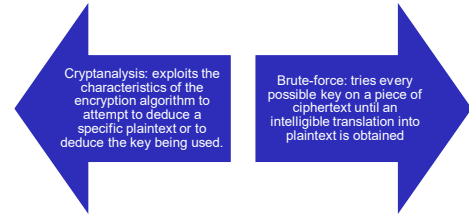
- Respective algorithms

- In the past, based on alphabet shifts (Cesar's Shiffre)
- More elaborate schemes applied today
 - Provide for confidentiality, integrity, and partially privacy

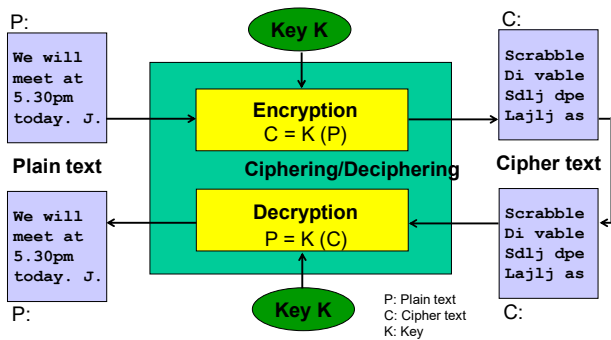


Types of Attack on Encrypted Messages

- Typically, the **objective** of attacking an encryption system is to **recover the key** in use rather than simply to recover the plaintext of a single ciphertext
 - Two major approaches



Basic Principle of Cryptography



Crypto Systems (1)

- **Symmetric cryptography**

- Entities own a shared, secret key



- **Asymmetric cryptography** (public key cryptography)

- Key pair of private/public parts



- Advantages

- Easy to understand/use
- Small overhead/calculation

- Drawbacks

- Key exchange complicated
- Commitment is “hard” (shared)

- Advantages

- Public keys easy to publish
- Key pair(s) per participant, no sharing

- Drawbacks

- Longer keys
- Larger overhead/calculation

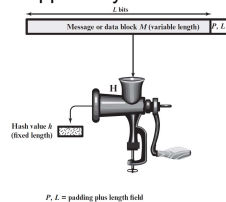
Crypto Systems (2)

- **Hash functions**

- Does not involve the use of any keys and it is used to check integrity of messages
- There are good and bad hash functions. Good ones produce outputs that are evenly distributed and apparently random.

- Defined as a function $h = H(M)$

- Variable input length of M bits
- Fixed output length of L bits
- **Irreversible**: from L is not possible to achieve the input M
- **Verifiable**: a single M will always produce the same L



P, L = padding plus length field

SYMMETRIC ENCRYPTION

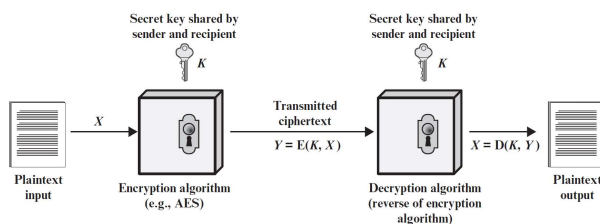
Cipher Model

- Also referred to as conventional or single-key encryption
- A symmetric encryption scheme has five ingredients:
 1. **Plaintext**: This is the original intelligible message or data that is fed into the algorithm as input.
 2. **Encryption algorithm**: The encryption algorithm performs various substitutions and transformations on the plaintext.
 3. **Secret key**: also an input to the encryption algorithm. The key is a value independent of the plaintext and algorithm.
 4. **Ciphertext**: the scrambled message produced as output
 5. **Decryption algorithm**: encryption algorithm run in reverse

Secure Use Requirements

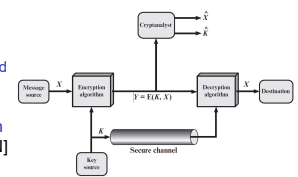
- There are two requirements for secure use of conventional encryption
 1. **A strong encryption algorithm**. The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext
 2. **Sender and receiver must have obtained copies of the secret key in a secure fashion** and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable

Simplified Model of Symmetric Encryption



Model of Symmetric Cryptosystem

- Source produces message X in plaintext based on a finite alphabet
 - $X = [X_1, X_2, \dots, X_M]$
- Based on a **shared key K in form $K = [K_1, K_2, \dots, K_J]$**
 - Secretly shared between source/recipient or provided by a trusted third-party
- Based on message X and encryption key K as input, the **encryption algorithm forms the ciphertext $Y = [Y_1, Y_2, \dots, Y_N]$**
 - Written as $Y = E(K, X)$
- **Recipient**, in the possession of the key K , is able to invert the transformation
 - Written as $X = D(K, Y)$
- **Opponents** observe Y but **do not have access to K or X** .



Symmetric Techniques

- **Substitution**
 - A letter is replaced by another letter, number, or symbols
- **Transposition**
 - A variation of substitution where plaintext is transposed in according to an algorithm in a finite alphabet
- **Rotor machines**
 - Based on electromechanical flow coding used to encrypt and decrypt messages.
 - The Enigma machine used 3 rotors
- **Steganography**
 - Techniques to hide the existence of one message within another, a form of security by obscurantism



Substitution (1)

- **Caesar Cipher**
 - Involves replacing each letter of the alphabet with the letter standing three places ahead in the alphabet
- plain: meet me after the toga party
cipher: PHHW PH DIWHU WKH WRJD SDUWB
- If you assign a numerical equivalent to each symbol:
- | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| a | b | c | d | e | f | g | h | i | j | k | l | m |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| n | o | p | q | r | s | t | u | v | w | x | y | z |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
- Then you can express the encryption algorithm as:

$$C = E(3, p) = (p + 3) \bmod 26 \quad \rightarrow \text{Based on a shift 3}$$

$$C = E(3, p) = (p + k) \bmod 26 \quad \rightarrow \text{Based on a shift } k$$

Substitution (2)

- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)

Dimensions	DES	AES
Developed	1976	1999
Key Length	56 bits	128, 192, 256 bits
Cypher Type	Block	Block
Plaintext Size	64 bits	128, 192, 256 bits
Rounds	16 rounds	10 rounds at 128 bits 12 rounds at 192 bits 14 rounds at 256 bits
Round names	Expansion Permutation, Xor, S-box, P-box, Xor and Swap.	Subbytes, Shiftrows, Mix columns, Addroundkeys.
Security	Insecure as of today, small key-size	Considered secure as of today

© 2024 UZH, CSG@ifi

19

ifi

Transposition

- Rail fence

Plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

m e m a t r h t g p r y e t e f e t e o a a t ⇒ MEMATRHTGPRYETEFETEOAAT

A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns.

```

Key:      4 3 1 2 5 6 7
Plaintext: a t t a c k p
           o s t p o n e
           d u n t i l t
           w o a m x y z

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ
    
```

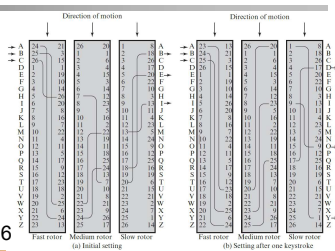
© 2024 UZH, CSG@ifi

20

ifi

Rotor Machines

- Consists of a set of independently rotating cylinders through which electrical pulses can flow. Substitutions:
 - 1 cylinder → 26
 - 2 cylinders → 26 x 26
 - 3 cylinders → 26 x 26 x 26



- Enigma



Classic version with 3 rotors
Rare version with 8 rotors

- Setting with 3 rotors with an alphabet of 26 letters
 - For every complete rotation of the inner cylinder, the middle cylinder rotates one pin position

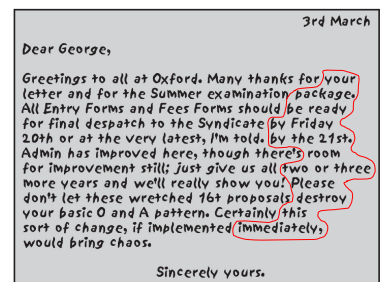
© 2024 UZH, CSG@ifi

21

ifi

Steganography

- Not an encryption method but only hiding of information
- Drawbacks
 - Requires a lot of overhead to hide a relatively few bits of information
 - Once discovered, it is worthless
- Many other forms
 - Invisible ink, character marking, pin punctures



© 2024 UZH, CSG@ifi

22

ifi

Stream and Block Ciphers (1)

- Block cipher: encrypt information in chunks by breaking plaintext into fixed-size blocks
 - Examples: DES, AES
- Advantages
 - High diffusion: information from one plaintext symbol is diffused into several ciphertext symbols.
 - Immunity to tampering: difficult to insert symbols without detection.
- Disadvantages
 - Slowness of encryption: an entire block must be accumulated before encryption / decryption can begin.
 - Error propagation: An error in one symbol may corrupt the entire block

© 2024 UZH, CSG@ifi

23

ifi

Stream and Block Ciphers (2)

- Stream cipher: encrypt plaintext information based on their single bits, which are then converted individually into ciphertext using key bits
 - Examples: Rivest Cipher (RC4), Salsa20
- Advantages
 - Speed of transformation: algorithms are linear in time and constant in space.
 - Low error propagation: an error in encrypting one symbol likely will not affect subsequent symbols.
- Disadvantages
 - Low diffusion: all information of a plaintext symbol is contained in a single ciphertext symbol.
 - Susceptibility to insertions/ modifications: an active interceptor who breaks the algorithm might insert spurious text that looks authentic

© 2024 UZH, CSG@ifi

24

ifi

ASYMMETRIC ENCRYPTION

Definitions Asymmetric Encryption

- NIST Glossary of Key Information Security Terms, NIST IR 7298
 - **Asymmetric Keys**: Two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.
 - **Public Key Certificate**: A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key. The certificate indicates that the subscriber identified in the certificate has sole control and access to the corresponding private key.
 - **Public Key (Asymmetric) Cryptographic Algorithm**: A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible.
 - **Public Key Infrastructure (PKI)**: A set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue, maintain, and revoke public key certificates.

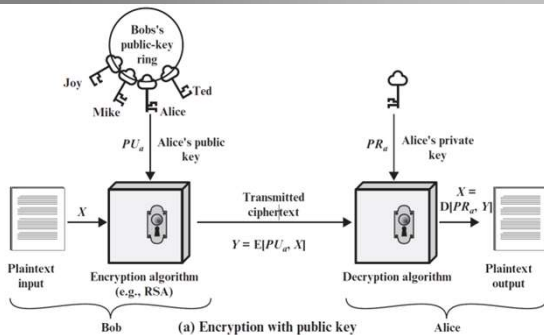
Public-key Cryptography Model

- A public-key encryption scheme has five ingredients
 1. **Plaintext**: The readable message or data that is fed into the algorithm as input.
 2. **Encryption algorithm**: The encryption algorithm performs various transformations on the plaintext.
 3. **Decryption algorithm**: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.
 4. **Public and private keys**: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.
 5. **Ciphertext**: This is the encrypted message produced as output. It depends on the plaintext and the key.

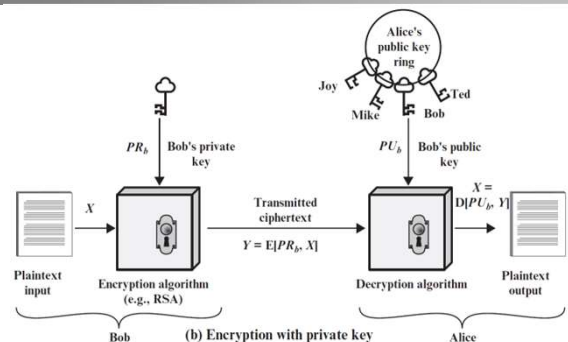
Requirements

- Diffie and Hellman postulated four conditions that such Public-Key encryption algorithms must fulfill
 1. It is **computationally easy** for a party B **to generate a key pair** (public key PUB, private key PRB)
 2. It is **computationally easy for a sender A**, knowing the public key and the message to be encrypted, M, **to generate the corresponding ciphertext**: $C = E(PUB, M)$
 3. It is **computationally easy for the receiver B to decrypt the resulting ciphertext** using the private key to recover the original message: $M = D(PR_b, C) = D(PR_b, E(PUB, M))$
 4. It is **computationally infeasible for an adversary**, knowing the public key, PUB, **to determine the private key**, PRB

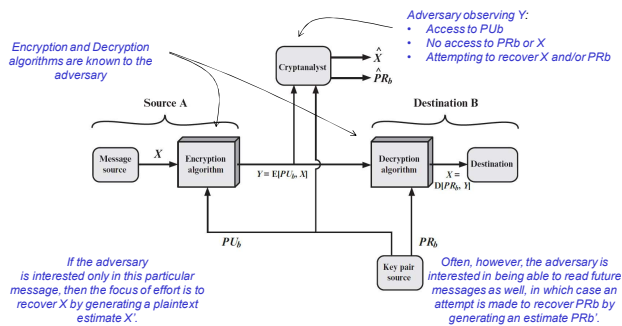
Encryption with Public Key



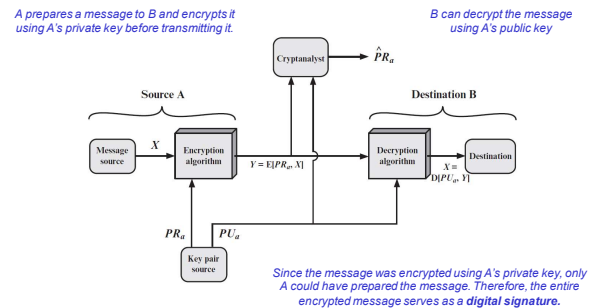
Encryption with Private Key



Adversaries: Confidentiality



Adversaries: Authentication



Asymmetric Encryption

- Asymmetric encryption (Public Key Encryption)
 - Encryption easy and publicly accessible to everyone
 - Decryption difficult for everyone except the intended recipient
 - Minimum key length 1,024 bit (309 decimal digits)
 - Secure algorithms: RSA (Rivest-Shamir-Adelman), ElGamal, and ECC (Elliptic Curve Cryptography)
- Practical encryption: Hybrid approaches
 - First: User authentication and exchange of a session key, public-key-based (in a non-hybrid version: symmetric)
 - Second: Symmetric encryption of user data by session key and further authentication required with session key
 - Note: Longer sessions should change session key on a periodical basis, e.g., once every 30 min or 1 hour

RSA (1)

- Encryption defines a function f , mapping plain text to cipher text; decryption defines the inverse of f
- f requires 5 properties
 - f is one-to-one (uniquely invertible)
 - f is easy to compute (encryption easy)
 - f^{-1} is difficult to compute (decryption difficult for senders)
 - f has a domain that is easy to sample from (Bob easily generates a key)
 - Existence of an easy-to-compute function d of the input of f making computing f^{-1} easy (Alice decrypts easily)

RSA: Rivest-Shamir-Adelman

f is one-way
is a trapdoor function

RSA (2)

- Example trapdoor function
 - $f(x, e, p, q) = (x^e \bmod pq, pq, e)$
 - p and q are prime numbers, x (encoded) plaintext, $y = x^e \bmod pq$ is the cipher text
 - $d(x, e, p, q) = e^{-1} \bmod (p-1) \cdot (q-1)$ (property 5.) that is $ed \bmod (p-1) \cdot (q-1) = 1$
- f is the basis for the RSA Cryptosystem
 - (e, pq) public encryption key (public key)
 - (p, q, d) secret decryption key (secret key)
 - Alice generates a private/public key pair once (re-use later)
 - Alice encrypts (or signs) message with her private key
 - Bob decrypts this message with Alice's public key

Simplified RSA "Calculation" (1)

- 2 (large, very large) prime numbers p and q are selected
 - $p = 11, q = 13$
- The product N is calculated as $N = p \cdot q = 143$
- Value e is calculated to: $1 < e < (p-1) \cdot (q-1)$
 - $e = 23$ (one possible value satisfying this inequality)
- Value d is calculated to: $d \cdot e \bmod (p-1) \cdot (q-1) = 1$
 - $d = 47$
- Thus, the number pair (e, N) defines the public key
 - $K_{pub} = (23, 143)$
- Thus, the triple of (p, q, d) defines the private key
 - $K_{priv} = (11, 13, 47)$

Simplified RSA “Calculation” (2)

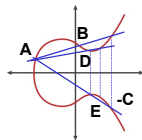
- Text (plain text) to be decrypted: $P = 75$
- Calculation of the **cipher text** y as: $y = P^e \bmod N$
 - $y = 75^{23} \bmod 143$
 - $y = 69$
- Cipher text $C = 69$ will be transmitted as the message
- Calculation of the **plain text** x as: $x = C^d \bmod N$
 - $x = 69^{47} \bmod 143$
 - $x = 75 \rightarrow P$

Theoretical Security Made Applicable

- Basis: **Prime Number Factoring**
 - $N = p \cdot q$, with p and q prime numbers
 - Existing algorithms cannot factorize N in “finite” time
 - 200 digit binary number factorization took from December 2003 until May 2005 at University of Bonn with 80 2.2 GHz machines
 - Typical key length: 300 binary digits
 - RSA Laboratories: Contest RSA 576 bit digits and RSA-2048 (617 decimal digits), RSA-768, 3 years, 2000 2.2GHz (Dec 19, 2009)
 - Exponential algorithms in place, but no polynomial ones? No proof!
 - But: Shor-Algorithm for quantum computers factorizes in polynomial time
 - Moore's Law (duplication of processing power in 2 years) does not determine a problem for RSA, however, quantum computers may simply break the system once eventually

Basic Elliptic Curve Cryptography (ECC)

- ECC generates a public-private key pair
 - A 256 bit key in ECC is at about “the same” security as a 3,072 bit key using RSA
 - ECC trapdoor function is similar to a mathematical “game of pool”
 - Knowing where starting point (A) is and how many hops are required to get to the ending point (E), it is “easy” to find the ending point
 - But by knowing where the starting point/ending points are, it is nearly impossible to find how many hops it took to get there
- Public Key: Starting Point A, Ending Point E
- Private Key: Number of hops from A to E

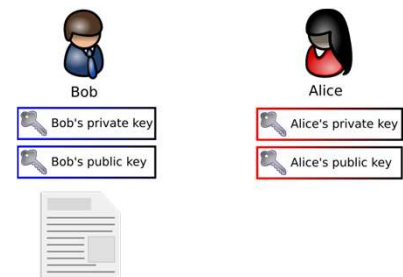


<https://blog.godtaudience.com/very-basic-elliptic-curve-cryptography-16c4fbc349ed>

Asymmetric Encryption Applied (1)

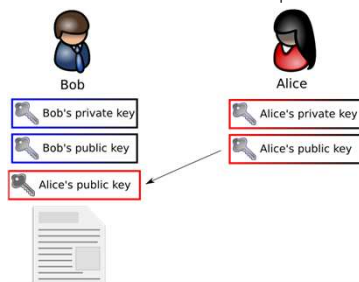
- The demand for **encryption**
 - Bob plans to send Alice an encrypted document

- Initial **set-up**



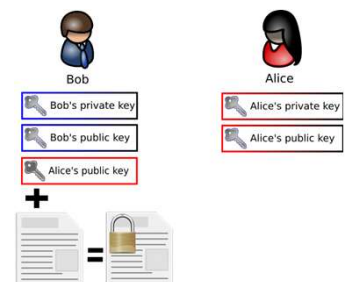
Asymmetric Encryption Applied (2)

- The **key exchange**
 - Alice hands over to Bob her public key ($K_{\text{Alice}_{\text{pub}}}$)
- Alternative
 - Bob may access Alice's public key in a key data base, too



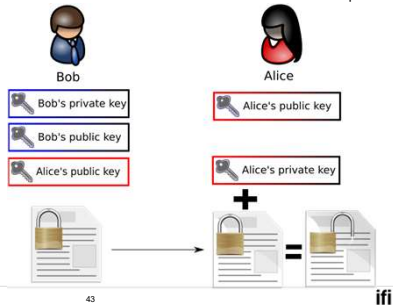
Asymmetric Encryption Applied (3)

- The **encryption**
 - Bob encrypts the document with Alice's public key ($K_{\text{Alice}_{\text{pub}}}$)



Asymmetric Encryption Applied (4)

- The **decryption**
 - Alice decrypts the document with her private key ($K_{\text{Alice}_{\text{priv}}}$)



© 2024 UZH, CSG@ifi

43

ifi

Summary Applications (1)

- In broad terms, we can classify the use of **public-key cryptosystems** into three categories
 - **Encryption/decryption**: The sender encrypts a message with the recipient's public key, and the recipient decrypts the message with the recipient's private key.
 - **Digital signature**: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
 - **Key exchange**: Two sides cooperate to exchange a session key, which is a secret key for symmetric encryption generated for use for a particular transaction (or session) and valid for a short period of time.

© 2024 UZH, CSG@ifi

44

ifi

Summary Applications (2)

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

© 2024 UZH, CSG@ifi

45

DSS (Digital Signature Standard)

ifi

Symmetric vs Asymmetric

Symmetric

- **Needed to work**
 1. Same algorithm and key used for encryption and decryption.
 2. Sender and receiver must share the key.
- **Needed for security**
 1. Key must be kept secret.
 2. It must be impossible or impractical to decipher a message if the key is kept secret.
 3. Knowledge of the algorithm plus ciphertext must be insufficient to determine the key.

Asymmetric

- **Needed to work**
 1. One algorithm is used for encryption and a related for decryption with a pair of keys, one for encryption and one for decryption.
 2. Sender and receiver must each have one of the matched pair of keys (not the same one).
- **Needed for security**
 1. One of the two keys must be kept secret.
 2. It must be impossible or impractical to decipher a message if one of the keys is kept secret.
 3. Knowledge of the algorithm plus one of the keys plus ciphertext must be insufficient to determine the other key.

© 2024 UZH, CSG@ifi

46

ifi

CERTIFICATES AND DIGITAL SIGNATURES

© 2024 UZH, CSG@ifi

47

ifi

Certificates and Signatures (1)

- **Certificates**
 - To certify the association between a public key and a person
 - Certification Authorities (CA) release certificates
 - CAs typically organized hierarchically, they form a TTP
 - To authenticate a sender for his/her signing process
 - Trusted mechanism to relate signature and signee
 - Applied to secure the signature, not the confidentiality of the data itself
- **Signatures**
 - Digital signatures (cf. below)
 - Achieved by asymmetric encryption
 - RSA
 - DSA (Digital Signature Algorithm)

© 2024 UZH, CSG@ifi

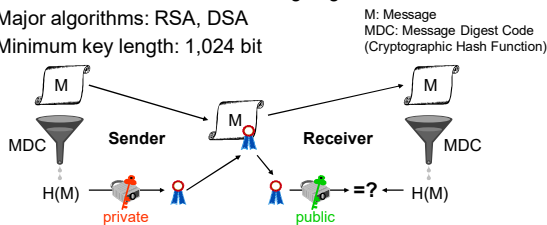
48

ifi

Certificates and Signatures (2)

□ Digital signature

- Hash value $H(M)$ signed with the private key of the sender
 - Receivers can check signature with the public key of the sender
- Guarantees the commitment of signing
- Major algorithms: RSA, DSA
- Minimum key length: 1,024 bit



© 2024 UZH, CSG@ifi

49

ifi

Digital Signature Applied (1)

□ Applying a digital signature

- Bob plans to send a digitally signed document M to Alice

□ Initial set-up



© 2024 UZH, CSG@ifi

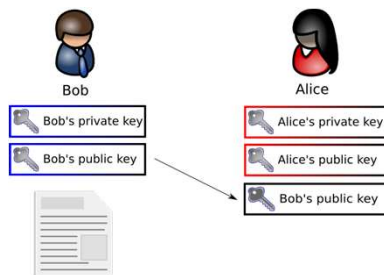
50

ifi

Digital Signature Applied (2)

□ The key exchange

- Bob hands over to Alice his public key ($K\text{-}Bob_{pub}$)



© 2024 UZH, CSG@ifi

51

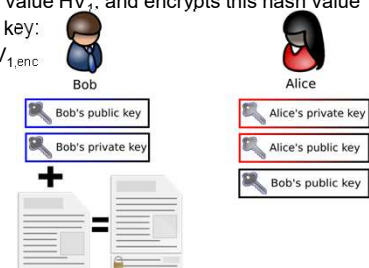
ifi

Digital Signature Applied (3)

□ Bob signs the document

- Bob applies a cryptographic hash function H to the document M , obtaining a hash value HV_1 , and encrypts this hash value HV_1 with his private key:

$$K\text{-}Bob_{priv}(HV_1) = HV_{1,enc}$$



© 2024 UZH, CSG@ifi

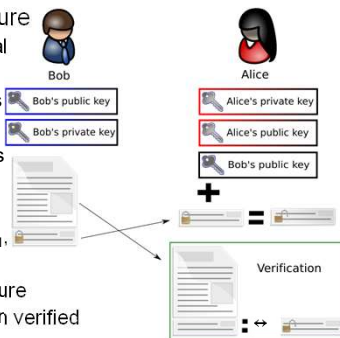
52

ifi

Digital Signature Applied (4)

□ Alice verifies the signature

1. Alice applies the identical hash function H onto the document M and obtains the hash value HV_2
2. Alice decrypts with Bob's public key the encrypted hash value $HV_{1,enc}$:
 $K\text{-}Bob_{pub}(HV_{1,enc}) = HV_1$,
 and obtains HV_1
3. If $HV_1 \leftrightarrow HV_2$, the signature on document M has been verified



© 2024 UZH, CSG@ifi

53

ifi

Example Digital Signature in PGP

□ Signed e-mail via the public domain tool PGP (Pretty Good Privacy)

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Dieses Mail ist signiert.
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.6 (GNU/Linux)

iD8DBQFHYACBQvL3WYjGY/sRAuyBAJ0UsM/rdXyrqTez79gybb
RStb0SJgCg0xb+dqOTAsRvM806wld7V+FPTrM==oxnW
-----END PGP SIGNATURE-----
```

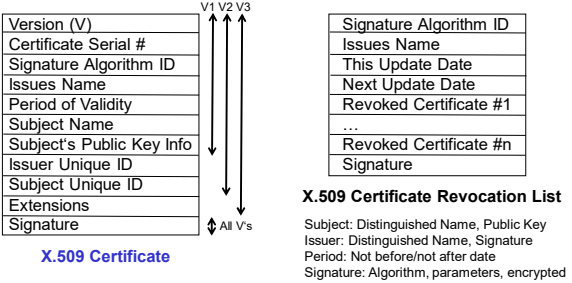
© 2024 UZH, CSG@ifi

54

ifi

Standard for Certificates

- ITU-T X.509 Certificates
 - International Telecommunications Union – Telecom Sector



Major Takeaways

- Symmetric and Asymmetric cryptography are the basis to ensure two major pillars of network security
 - Confidentiality and Integrity
- In terms of speed:
 - Advantage of symmetric cryptography has an advantage due to usage of a single key with smaller key-size
- In terms of security:
 - Advantage of asymmetric cryptography once keys does not need to be shared
- Security of several real-world applications relies on existing symmetric and asymmetric algorithms
 - Banking, data storage, digital certificates, blockchain

Computer Network and Security Principles (CNSP)

M03 – Layered Security (Data Link, Network, Transport, Application)

© 2024 UZH, CSG@ifi

1

ifi

Layered Security

- Objectives
 - To provide understand the OSI model and its layers
 - To provide a broad view on security issues and mechanisms applied to each layer
 - To overview selected protocols and examples
- Topics
 - Overview of the OSI model and comparison with TCP/IP model
 - General overview of security by network layers
 - Specific examples of protocols and mechanisms for each layer
 - Discussion and Takeaways

© 2024 UZH, CSG@ifi

2

ifi

Internet Evolution

□ Internet evolution

- 1960: ARPA, ARPANET
- 1970: TCP/IP
- 1980: DNS, E-mails
- 1990: HTML
- 2000: Google (search engines)
- 2010: Cloud



□ Evolution over time through many “patches” and fixes

- Subnets
- Mobile IP, Secure IP (IPSec)
- Domain Name System (DNS)
- Network Address Translation (NAT)
- Carrier-grade NAT



© 2024 UZH, CSG@ifi

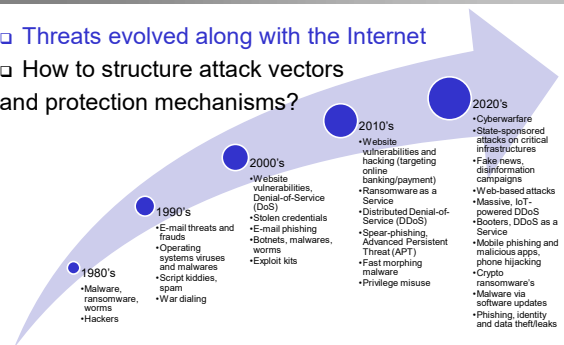
3

ifi

Internet Threat Evolution

□ Threats evolved along with the Internet

□ How to structure attack vectors and protection mechanisms?



© 2024 UZH, CSG@ifi

4

ifi

OVERVIEW OF THE OSI MODEL

© 2024 UZH, CSG@ifi

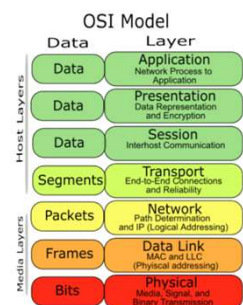
5

ifi

Open System Interconnection (OSI)

- OSI proposed as a theoretical model to define widely-accepted ways of setting up networks and connecting them together
 - Attempt to see hardware and software manufacturers to agree on framework to develop networking technologies
- Standardization by ISO began late 1970 and was widely accepted in 1980
 - Published in 1984
 - Not created for educational purposes, but evolved in this direction
 - Popularity rise of the Internet made the TCP/IP a de-facto standard
- Formal specification (behind a pay-wall)

Layers: 1 2 3 4 5 6 7



© 2024 UZH, CSG@ifi

6

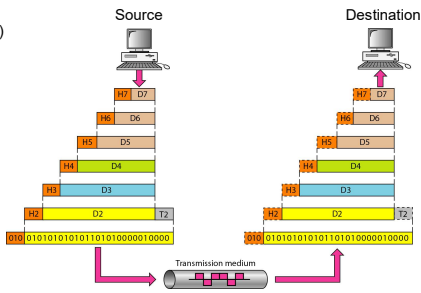
ifi

Figure: Edwards, P. N., Jackson, S. J., Bunker, G. C., & Kibbel, C. P. (2011). Understanding Infrastructure: Dynamics, Terminus, and design.

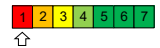
Encapsulation and Decapsulation using the OSI Model

Encapsulation

- A packet: data and header and maybe trailer)
- The data portion of a packet at level N-1 carries the whole packet from level N
- Level N-1 is not aware of which part of the packet is data, header, or trailer
- For level N-1, the whole packet coming from level N is treated as one integral unit



Physical Layer



Roles

- Encoding and Signaling
- Physical Data Transmission
- Hardware Specifications
- Topology and Design

Data Type

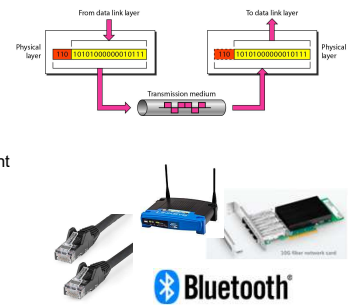
- Bits

Scope

- Electrical or light signals sent between local devices

Common Technologies

- Ethernet physical layer (10BASE-T, 100BASE-T)
- Bluetooth, 802.11
- ISDN, USB, PCI-E



Data Link Layer



Roles

- Logical Link Control (LLC)
- Media Access Control
- Data Framing**
- Addressing
- Error Detection and Handling
- Defining Requirements of Physical Layer

Data Type

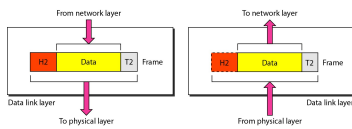
- Frames

Scope

- Low-level data messages between local devices

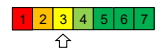
Common Protocols

- ARP, MPLS, Spanning Tree, Token Ring
- 802.11, Fiber Distributed Data Interface (FDDI)



	Frame	Packet
Layer	Data Link	Network
Addressing	Physical	Logical
Error Detection and Flow Control	✓	✗

Network Layer



Roles

- Logical Addressing
- Routing**
- Datagram Encapsulation
- Fragmentation and Reassembly
- Error Handling and Diagnostics

Data Type

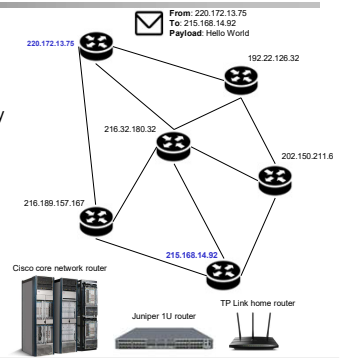
- Datagrams / Packets

Scope

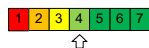
- Messages between local or remote devices

Common Protocols

- IP, IPv6, IP, NAT, IPsec
- Routing protocols such as RIP and BGP



Transport Layer



Roles

- Process-Level Addressing
- Multiplexing/Demultiplexing
- Connections
- Segmentation and Reassembly
- Acknowledgments and Retransmissions
- Flow Control

Data Type

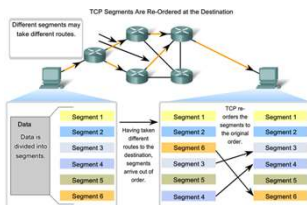
- Datagrams / Segments

Scope

- Communication between software processes

Common Protocols

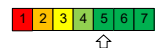
- TCP and UDP



Terminology:

- Segment:** If the transport protocol is TCP, the unit of data sent from TCP to network layer is called Segment.
- Datagram:** Term used at layer 3 and 4. At transport layer, if protocol is UDP, we use datagram. We differentiate them as UDP Datagram and IP Datagram

Session Layer



Roles

- Session Establishment, Management and Termination
- Authentication and Authorization
- Flow Control

Data Type

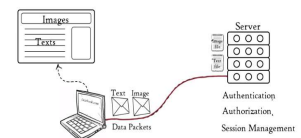
- Sessions

Scope

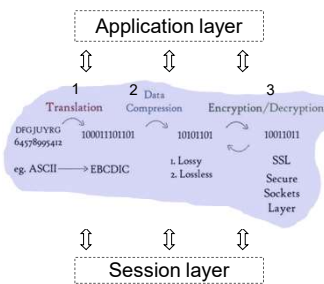
- Sessions between local or remote devices

Common Protocols

- SCP (Session Control Protocol)
- PPTP (Point-to-point Tunneling Protocol)
- PAP (Password Authentication Protocol)
- L2TP (Layer 2 Tunneling Protocol)
- NetBIOS



Presentation Layer



- Roles
 - Data Translation
 - Compression and Encryption
- Data Type
 - Encoded User Data
- Scope
 - Application data representations
- Common Protocols
 - SSL
 - Shells and Redirectors
 - MIME

Application Layer

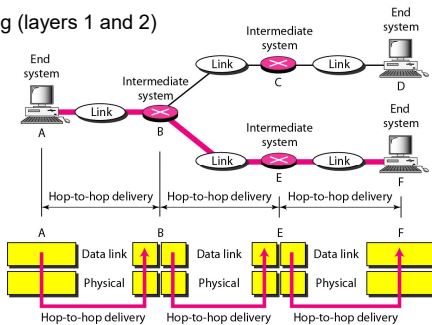


- Roles
 - User convenience, widely diverse application services
- Data Type
 - User Data
- Scope
 - Application data
- Common Protocols
 - DNS, DHCP, SNMP
 - RMON, FTP, TFTP,
 - SMTP, POP3, IMAP, HTTP, HTTPS



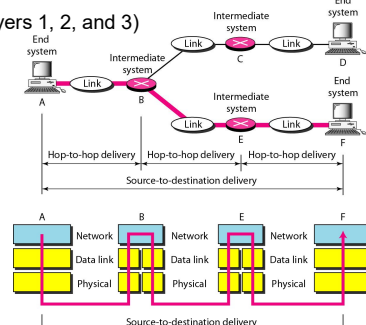
Hop-to-hop Delivery

Switching (layers 1 and 2)



Source-to-destination Delivery

Routing (layers 1, 2, and 3)



SECURITY OVERVIEW BY LAYERS

OSI Security X.800 and RFC 4949

- Attacks are classified as
 - Passive
 - Traffic analysis, packet sniffing, side-channel analysis
 - Active
 - Tampering data, replaying messages, denial-of-service, masquerade
- Security Services include (non-exhaustive)
 - Peer and data authentication
 - Access control
 - Confidentiality and Encryption
 - Data integrity
 - Availability
- Which attacks are applicable to which layers?
- Which security services are applicable to which layers?

Building Blocks for Network Security

- Encryption and authentication algorithms are building blocks of secure network protocols
- Deploying cryptographic algorithms at different layers have different security effects
- Where should we put the security protocol in the network architecture?

Security Overview by Layers

OSI Layers	Functions	Example Protocol	Example of Attacks
Application	User interface, process to application	HTTP, FTP, IMAP, SMTP	Malware, DoS, SMTP Attacks, Buffer Overflows
Presentation	Data format, encryption	SSL, TLS JPEG, MPEG	Phishing, session sniffing
Session	Process-to-process communication	SQL, RPC, NFS	Session Hijacking, Telnet and FTP sniffing
Transport	End-to-end communication maintenance	TCP, UDP	Desynchronization, TCP sequence prediction, SYN flooding, Denial-of-Service (DoS)
Network	Routing data, logical addresses, WAN	IP, ICMP	Sybil attacks, IP spoofing, port sniffing, Distributed Denial of Service (DDoS), Man-in-the-middle (MITM)
Data Link	Physical addressing, LAN	ARP	MAC Spoofing, Spanning Tree Attack, MAC/ARP sniffing
Physical	Transmission of bytes	802.11, Ethernet, Bluetooth	Sniffing (eavesdropping), Jamming signals, tampering data

OSI vs. TCP/IP

OSI	TCP/IP
Application Layer	Application Layer
Presentation Layer	
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	
Physical Layer	Link Layer

- Both are conceptual models
- While OSI is widely used as a theoretical reference, TCP/IP is closer to practical Internet implementations
- OSI is relatively more complex but define clear standards for each layer

TCP/IP Protocol Layers

- Logical (Software)
 - Application
 - Web, E-mail, DNS
 - Transport
 - TCP, UDP
 - Network Layer
 - IP, ICMP
- Physical (Hardware)
 - Data Link Layer
 - Ethernet, 802.11, ARP
 - Physical Layer
 - Wireless signals, physical security

SPECIFIC PROTOCOLS AND EXAMPLES

Physical Layer Security

- WiFi jamming makes use of **intentional radio interferences** to disrupt communication by keeping the medium **busy**
- Jamming leads to:
 - Network partition and disruption
 - Forcing packets to be routed over chosen paths
- Good applications:
 - Prison jammers, Airport drone jamming
- Bad applications:
 - Car key jamming, GPS tracker, jamming neighbors WiFi
- Strictly **forbidden** to own or import a Jammer device in Switzerland
 - Still... not difficult to build one based standard hardware (ESP32)
 - And there are many tutorials available



Data Link Layer Security (1)

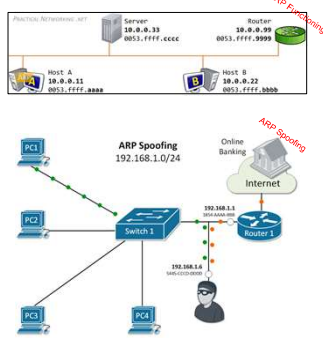
Address Resolution Protocol (ARP)

- Designed without security concerns in mind, i.e., no validation on ARP packets
- Map IP addresses to a physical machine address recognizable in the local Ethernet.

ARP Spoofing/Poisoning built on 2 principles:

- Goal:** intercept the victim's connection at the layer 2, typically forging the MAC address of the gateway.
- 1. Man-in-the-middle (MITM):
 - Active eavesdropping in which the attacker is "in the middle" of the victim's connection
- 2. Session Hijacking:
 - Attacker forge ARP responses to hijack the victim's connection
- Tools:** Ettercap, many repositories on GitHub, YouTube tutorials

End-to-end encryption (mainly on public networks)



Data Link Layer Security (2)

Advantages

- Transparent to applications
- Hardware solution possible
- Can address especially vulnerable links (viz. wireless)

Drawbacks

- Hop-by-hop protection causes multiple applications of crypto operations
- May not provide end to end security



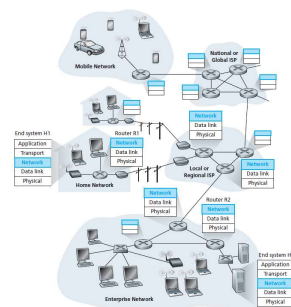
Network Layer Security

Suppose every link in our network had strong link-layer security

- Why would this not be enough?

We need security **across** networks. Ideally, **end-to-end**

At the network layer, this is usually accomplished with a Virtual Private Network (VPN)



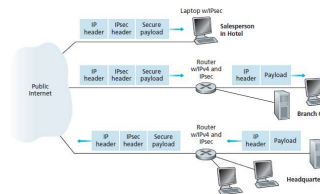
Virtual Private Networks (VPN)

Connect two (or more) networks

- that are physically isolated, and make them appear to be a single network
- Alternately: connect a single remote host (often a laptop) to one network

Goal: adversary between the networks should not be able to read or modify the traffic flowing across the VPN

- But DoS and some traffic analysis still usually possible



IPSec (1)

One standard way to set up a VPN is by using IPSec (Internet Protocol Security)

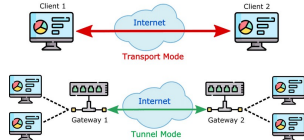
- Encrypt and authenticate traffic at the IP level

Three security functions

- Authentication
- Confidentiality
- Key management

Two modes

- Transport mode**
 - Useful for connecting a single laptop to a home network
 - Only the contents of the original IP packet are encrypted and authenticated
- Tunnel mode**
 - Useful for connecting two networks
 - The contents and the header of the original IP packet are encrypted and authenticated; result is placed inside a new IP packet destined for the remote VPN gateway



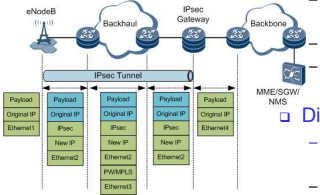
IPSec (2)

Advantages over application layer (TLS/SSL)

- Can be implemented at gateway and not only at the host/desktop
- Transparent to application programs and users
- Data can be sent unencrypted in LAN to avoid encryption overhead

Disadvantages of IPSec

- Performance** as the encryption/decryption of each packet requires processing power
- IPSec may be **restricted at firewalls**
- Increased complexity** once specification allows for a wide range of options to be configured
 - Issues at implementation or configuration



OpenVPN vs. WireGuard

OpenVPN



- Open-source
- Considered industry-standard
 - Codebase: around 70k lines
- Relatively **slower**
- Year released: 2001
- Authentication:
 - HMAC, X509 PKI
- Ciphers:
 - AES-256, AES-128, ChaCha20, Poly1305

WireGuard



- Open-source
- Lightweight, designed to be simple
 - Codebase: around 4k lines
- Relatively **faster**
- Year released: 2019
- Authentication:
 - Poly1305
- Ciphers:
 - ChaCha20-Poly1305, Curve25519, SipHash24, BLAKE2s

Network Layer Security

Advantages

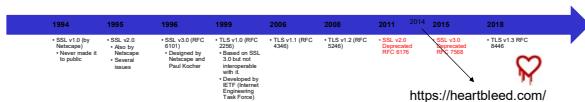
- Transparent to applications (upper layers)
- Provide confidentiality and integrity end-to-end
- Flexible approach

Drawbacks

- Makes routing more complex
 - Performance is often the biggest bottleneck
- Flexibility introduces policy management and compatibility challenges
 - Typically deployed in specific use-cases that require very secure connections

Transport Layer Security

- Goal is to establish a secure connection between source and destination
 - Heavily based on the PKI and CA to define a secure end-to-end connection
- Two major protocols
 - SSL (Secure Socket Layer) and its evolution
 - TLS (Transport Layer Security)
- Evolution timeline



SSL/TLS Guarantees

De facto standard for Internet security

- “The primary goal of the TLS protocol is to provide privacy and data integrity between two communicating applications”

End-to-end secure communications in the presence of a network attacker

- Attacker completely owns the network: controls Wi-Fi, DNS, routers, his own websites, can listen to any packet, modify packets in transit, inject his own packets into the network

Scenario

- User reads emails at an Internet café connected via a rooted Wi-Fi access point to a dodgy ISP in a hostile region

SSL Overview

SSL runs on top of TCP

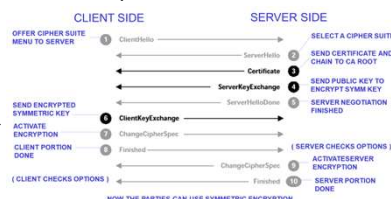
- Advantage: does not require changes to TCP
- Possibly the most important security-related ecommerce protocol

- From the programmer's viewpoint, it sits at the transport layer
 - TCP-based

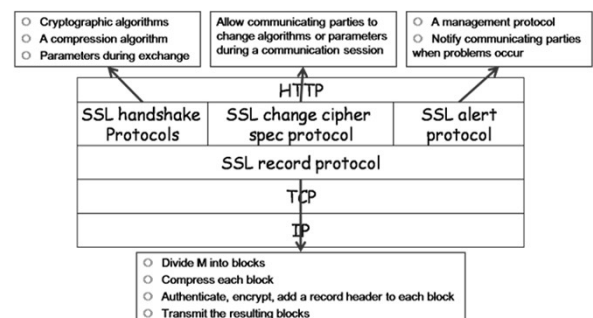
Basic SSL Functioning

Establish a session

- Agree on algorithms
- Share secrets
- Perform authentication
- Transfer application data
- Ensure privacy and integrity

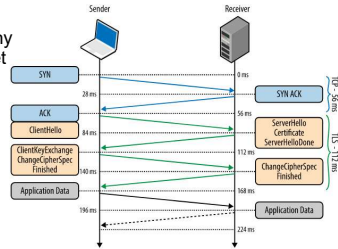


SSL Structure

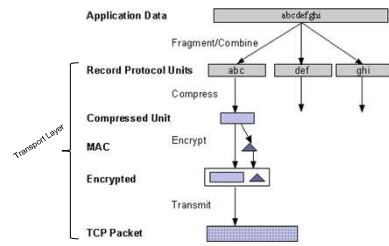


TLS Structure

- TLS consists of **two** layers of protocols
- **Handshake protocol**
 - Use public-key cryptography to establish a shared secret key between the client and the server
 - Similar to SSL
- **Record protocol**
 - Use the secret key established in the handshake protocol to protect communication between the client and the server



SSL/TLS Record Protocol



The **Record Protocol** is used to transfer application and SSL/TLS Control data between the client and server, where necessary fragmenting this data into smaller units, or combining multiple higher level protocol data messages into single units.

It may:

- Compress
- Attach digest signatures, and
- Encrypt these units before transmitting

*MAC = Message Authentication Code, piece of information (string) that helps to authenticate a message
*HMAC = Hashed-based MAC, a specific type of MAC using hash algorithms

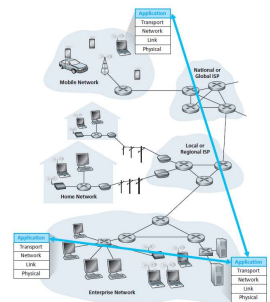
Transport Layer Security

- **Advantages**
 - Transparent to applications and may be packaged with applications
 - Exposing TCP enables compression and QoS classification
- **Disadvantages**
 - Probably implemented in software
 - Exposing TCP risks DoS



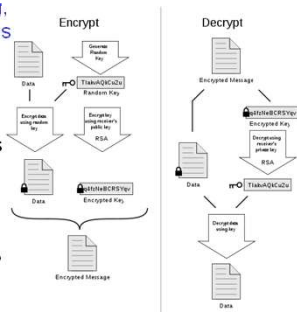
Application Layer Security

- Application **programs** (e.g., Netscape) use a set of well-defined application layer **protocols** (e.g., HTTP) and **formats** (e.g., HTML)
 - A given Application Layer protocol uses TCP or UDP



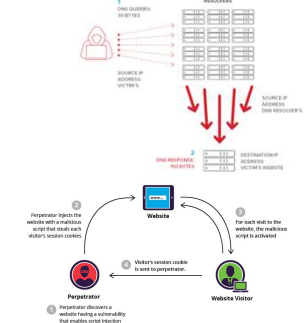
PGP (Pretty Good Privacy)

- PGP protocol used for **encrypting, decrypting, and signing messages** or files using a key pair
 - Application layer
 - To encrypt emails and files
 - Released in 1991
- **OpenPGP**: Pretty Good Privacy's open-source version
 - Universally-accepted PGP standard
 - Released in 1997
- **GPG**: GNU Privacy Guard
 - Popular solution following OpenPGP standards
 - Released in 1999



Further Application Layer Threats

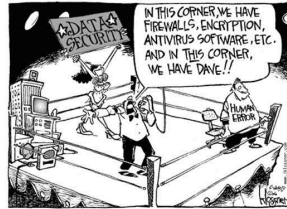
- DDoS attacks, which require applications to be shielded.
 - HTTP floods that aim to lock up applications and deny access to legitimate users.
 - **DNS amplification**
- SQL injections on applications that have poorly verified user input protocols.
- **Cross-site scripting (XSS)** and parameter tampering.
 - Injection of malicious scripts
 - <https://owasp.org/www-community/attacks/xss/>



Application Layer Security

Advantages

- Customized to application
- Requires no special protocol stack (transparent to networking)



Drawbacks

- Hard to share between applications (especially due to standardization challenges)

DISCUSSION AND TAKEAWAYS

Pros and Cons (1)

Application Layer

- Provides end-to-end security protection
- No need to decrypt data or check for signatures
- Attackers may analyze traffic and modify headers

Transport Layer

- Provides security protections for TCP packets
- No need to modify any application programs
- Attackers may analyze traffic via IP headers

Pros and Cons (2)

Network Layer

- Provides link-to-link security protection
- Transport mode: Encrypt payload only
- Tunnel mode: Encrypt both header & payload; need a gateway
- No need to modify any application programs

Data-link Layer

- Provides security protections for frames
- No need to modify any application programs
- Traffic analysis would not yield much info

Possible Placement of Security Services

Security Service	OSI Layer (X.800)						
	1	2	3	4	5	6	7
Peer entity authentication	x	x	✓	✓	x	x	✓
Data origin authentication	x	x	✓	✓	x	x	✓
Access control service	x	x	✓	✓	x	x	✓
Connection confidentiality	✓	✓	✓	✓	x	✓	✓
Connectionless confidentiality	x	✓	✓	✓	x	✓	✓
Selective field confidentiality	x	x	x	x	x	✓	✓
Traffic flow confidentiality	✓	x	✓	x	x	x	✓
Connection integrity with recovery	x	x	x	✓	x	x	✓
Connection integrity without recovery	x	x	✓	✓	x	x	✓
Selective field connectionless integrity	x	x	x	x	x	x	✓
Connectionless integrity	x	x	✓	✓	x	x	✓
Selective field connectionless integrity	x	x	x	x	x	x	✓
Non-repudiation of origin	x	x	x	x	x	x	✓
Non-repudiation of delivery	x	x	x	x	x	x	✓

Major Takeaways

- Network protocols are designed in layers; **security can be provided at multiple layers** with various tradeoffs
- Security and the type of threats differ in **significant ways depending on higher layers**
 - Higher diversity of protocols and applications
- Important differences between **theoretical descriptions, standards, and operational software**
 - Evolution (versions, extensibility)
 - Interoperability (options, negotiation)
 - Error modes (subset handled only)

Computer Network and Security Principles (CNSP)

L04 – Cloud Security

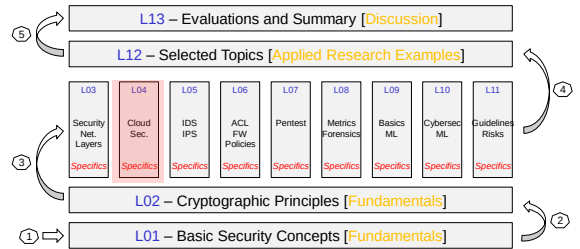
© 2024 UZH, CSG@ifi

1

ifi

1

Overview of Class and Lectures (Chapter Dependency)



© 2024 UZH, CSG@ifi

2

ifi

2

Cloud Security

- Objectives
 - To understand the basics of cloud computing
 - To provide a broad view on security models and services
 - To overview top threats associated to cloud computing
 - To overview major security guidelines and security taxonomy
- Topics
 - Cloud fundamentals and service models. Including serverless computing
 - Basic risks in the adoption, portability, and migration of cloud services
 - Detailed overview of major threats in cloud computing
 - General security guidelines as well as virtualization guidelines
 - Summary of recommendations and takeaways

© 2024 UZH, CSG@ifi

3

ifi

3

CLOUD FUNDAMENTALS

© 2024 UZH, CSG@ifi

4

ifi

4

From Desktops to Clouds

- Desktops and workstations
 - Networked, low-capacity storage, limited processing
- Local clusters
 - Connected, centralized, higher storage/processing
- Institutional clusters, High-performance Computing HPC
 - Access-controlled, centralized, high storage/processing
- Grids
 - Shared services on top of visible hardware and software resources
- Clouds
 - Big data storage and computing



Great Globular Cluster in Hercules

© 2024 UZH, CSG@ifi

5

<https://www.top500.org/lists/top500/2023/06/>

ifi

5

Clouds – History (1)

- Clouds driven by business-oriented and commercial services
- History – big players
 - Salesforce in 1999 (CRM products, SaaS)
 - Enterprise application via Web/API
 - Amazon 2002
 - Web services, AWS "Developers can now incorporate Amazon.com content and features directly onto their own websites. We can't wait to see how they're going to surprise us." - Jeff Bezos, founder and CEO of Amazon.com
 - In 2006 EC2 – rent virtual computers (S, M, L, XL)

<https://www.salesforce.com/eu/?ir=1>



© 2024 UZH, CSG@ifi

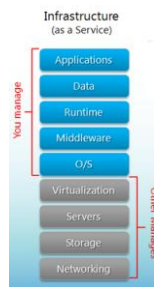
6

ifi

6

IaaS

- For computing, provides an infrastructure (virtual machine, storage, firewall, load balancer)
 - Load an OS image
 - Maintenance of the infrastructure (applying security patches) not done by cloud provider
- E.g., Amazon EC2 – purchasing options
 - On-Demand Instances (pay per hour, no commitment)
 - Reserved Instances (pay one year, commitment, lower hourly rate)
 - Spot Instances (get instance when price is right)
 - No time-sensitive tasks



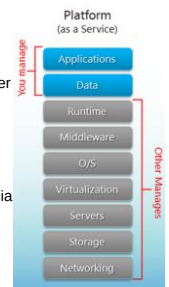
© 2024 UZH, CSG@#

13

ifi

PaaS

- Amazon Web Services (AWS), Google Cloud Platform (AppEngine), MS commercial cloud (Azure), Oracle
- PaaS provides an API
 - Uses tools and libraries provided by the PaaS provider
 - No security patches for infrastructure
 - Difficult to migrate from one PaaS vendor to another
- Example AppEngine
 - Python, Java*, Go, runs Spring/Struts/Grails/Django
 - 99.95% uptime SLA – if not met, user receives financial credit
- AppEngine PaaS vs. IaaS
 - AppEngine better scalable, but application tailored
 - IaaS, install your own OS, "Quadraple Extra Large" → then scale on your own



© 2024 UZH, CSG@#

14

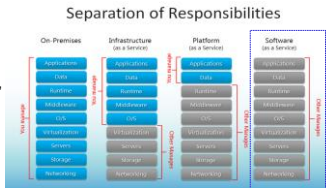
ifi

13

14

SaaS

- Software-as-a-Service
 - One application, e.g., Email/content management/CRM
 - Many users, typically browser-based, e.g., Gmail, Dropbox, iCloud, ...
 - Free or subscription based, pay for usage
 - Integration protocols
 - Typically, HTTP, REST, SOAP and JSON
- IaaS – PaaS – SaaS
 - Who manages what?



<http://blogs.technet.com/b/veinremde/archive/2011/04/03/saas-paas-and-iaas-oh-my-quot-cloudy-april-quot-part-3.aspx>

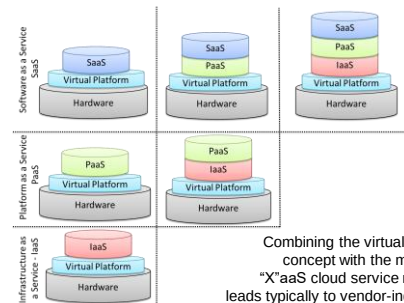
© 2024 UZH, CSG@#

15

ifi

15

Flexibility of Combinations



Combining the virtualization concept with the main "X"aaS cloud service models leads typically to vendor-independency

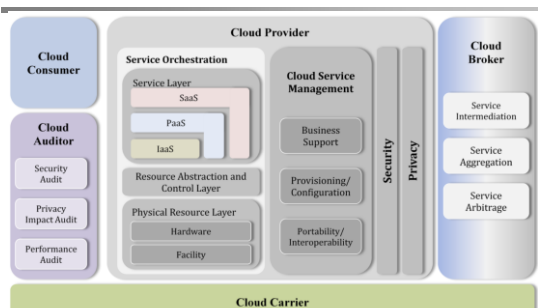
© 2024 UZH, CSG@#

16

ifi

16

Cloud Reference Model



© 2024 UZH, CSG@#

17

NIST Cloud Computing Reference Architecture, Special Publication 800-202
URL: <https://nvlpubs.nist.gov/nistpubs/specialpubs/SP800-202.pdf>

ifi

17

Serverless Computing

- Increased demand for software agility and delivery
 - Faster deployment (Dev) and operation (Ops) of cloud services
 - A "DevOps Engineer" bridges the gap between the Dev and operations teams, offering maintenance on-the-fly
- Two common terms/tasks define DevOps
 - Continuous Development (CD): short cycles of software update
 - Continuous Integration (CI): automated build and testing of new code at every update cycle
- Direct impact on how cloud services are provided
 - Serverless computing: rethinking how to deploy and run software in cloud

© 2024 UZH, CSG@#

18

ifi

18

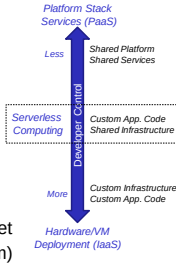
Defining “Serverless Computing”

- Relatively new concept emerging from industry
 - No single, formal definition in place yet

“Serverless computing is a method of providing backend services on an as-used basis. Servers are still used, but a company that gets backend services from a serverless vendor is charged based on usage, not a fixed amount of bandwidth or number of servers.”

“Serverless technologies feature automatic scaling, built-in high availability, and a pay-for-use billing model to increase agility and optimize costs. These technologies also eliminate infrastructure management tasks like capacity provisioning and patching, so you can focus on writing code that serves your customers.”

“Serverless computing enables developers to build applications faster by eliminating the need for them to manage infrastructure. With serverless applications, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code.”



– **Function-as-a-Service (FaaS)** is a subset of serverless (central computing paradigm)

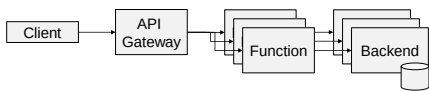
© 2024 UZH, CSG@UZH

19

ifi

19

General Serverless Architecture



API Gateway: Creates, publishes secure APIs

- Acting as a router for applications access data and functionalities.
- Requests are authenticated
- Accepts and processes up thousands of concurrent API calls

Function: Defines the logic to run backend code

- Web applications, IoT and mobile backends
- Cost based on the compute time consumed
- Monolithic or microservices

Backend: Handles compute and storage requests of functions

- On-demand capacity
- Key-value, documents, time-series databases

Key characteristics

- No operating system to configure or manage
- Ability to scale automatically with the usage
- Built-in availability and fault tolerance, automatically managed by the cloud provider
- Pay-per-invocation billing model (most cases)



© 2024 UZH, CSG@UZH

20

ifi

20

FaaS Discussion

- **Advantages**
 - **Reduced time** to deploy and run applications
 - **Flexibility** to adapt and evolve a running application by adding more functions
 - **Multiple (and simultaneous)** programming languages and databases
 - **Modularity**
 - Application logic may be mapped in independent functions
- **Drawbacks**
 - Code **overhead** (repeated code)
 - Increased **complexity**
 - **Cold starts**
 - Slower general **response time**
 - Code **not portable** (lock-in)
 - **Data security**



© 2024 UZH, CSG@UZH

21

ifi

21

Serverless Future?



- **Relatively new technology**
 - Concept of FaaS started with Google App Engine in 2008
 - FaaS was rethought as “serverless computing” by AWS Lambda functions in 2015
- **Use cases are still unclear**
 - Proven useful to process IoT data streams (big data use cases), but benefits in traditional Web services still unclear
 - **Easy deployment of new functions**, but overall performance and code maintenance may be hindered
 - Example: experience with AWS might be cumbersome, since there is no precise error description
 - Sometimes deleting “everything” and redeploying fixes a bug, too

© 2024 UZH, CSG@UZH

22

ifi

22

Cloud Security

- At the end of the day, data is stored “somewhere”
 - Known, unknown, legally compliant?



© 2024 UZH, CSG@UZH

23

ifi

23

© 2024 UZH, CSG@UZH

24

ifi

24

SECURITY AND SERVICE MODELS



Cloud Adoption is Still Growing...

- The need to run a **business** more efficiently, reducing time-to-market, and enhancing user experience are still driving companies to embrace the cloud
 - Cf. IaaS, SaaS, PaaS, or FaaS (as discussed before)



© 2024 UZH, CSG@#

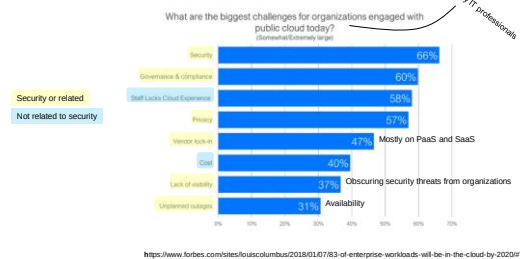
25

ifi

25

...but Security Remains a Concern

Public Cloud Challenges



© 2024 UZH, CSG@#

26

ifi

26

Basic Risks in Clouds (1)

National Institute of Standards and Technology: An Introduction to Computer Security: The NIST Handbook, 800-12.

- Risk = Threat X Vulnerabilities
 - Threat
 - Any potentially malicious or other occurrence that can have an undesirable effect on the assets and resources of the Cloud system
 - Vulnerability
 - A quality or characteristic of the Cloud-based system that provides an opportunity for misuse (internal or external)
 - Manifold **Cloud-based threats, vulnerabilities** (external)
 - Eavesdropping, changing data, DDoS, provider shut-down ...
 - Threats for confidentiality, integrity, authenticity, availability ...
 - Service interfaces, data transfer, data storage, virtual machines/sand-box, accounting data ...
- Risks are expressed typically in monetary values

© 2024 UZH, CSG@#

27

ifi

27

Basic Risks in Clouds (2)

- **Unauthorized access** to customer and business data
 - Using cloud services increases value of that service as a potential target
 - ⇒ Concentrates risk on a single point of failure
- **Security risks** at the vendor/provider side
 - E.g., storing critical data (customer payment data, mailing lists)
 - ⇒ Trust placed into the hands of another company
- **Compliance and legal risks**
 - Regulated data security: healthcare, banking, government, ...
 - ⇒ Where is the data stored? What if data breaches occur?
- Risks related to **lack of control**
 - No guarantee that features used today will be provided tomorrow, price
- **Availability risks**: Cloud provider and ISP are "external"
 - https://www.calyptix.com/research-2/top-5-risks-of-cloud-computing/

© 2024 UZH, CSG@#

28

ifi

28

Basic Risks in Clouds (3)

- Implications of cloud architecture on existing strategies
 - Traditional security models focus on **securing the perimeter** (e.g., firewalling) – internal devices are **trusted**
 - Mobile devices and cloud services lead to an open and "fuzzy" perimeter
 - Traditional perimeter-based security models **fail** to incorporate these devices and services under reasonable cost
 - Zero-trust security models
 - Always verify access ("no threat-free environments")
 - E.g., authenticate at every step
 - Limit impacts of a breach ("assume breach")
 - E.g., assign privileges with minimum capabilities to do the job

https://www.microsoft.com/security/blog/2019/10/23/perimeter-based-network-defense-transform-zero-trust-model/

© 2024 UZH, CSG@#

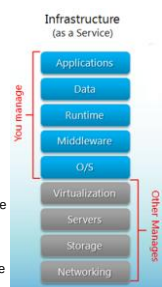
29

ifi

29

Cloud Security: A Key Dependability

- An **additional layer** to consider, virtualization
 - Must be properly configured, managed, secured
- **Security concerns**
 - Data breaches (not unique to cloud computing)
 - Human error (95% of failures will be the customer's fault)
 - Data loss with no backup
 - Insider threats (53% of organizations surveyed confirmed insider attacks against their organization)
 - DDoS (Distributed Denial-of-Service) attacks (lengthy service outages, reputational damage, exposure of customer data)
 - Insecure APIs (public "front door" to all applications)
 - Exploits (multi-tenancy nature of the cloud - customers share computing resources - means shared memory/resources)



© 2024 UZH, CSG@#

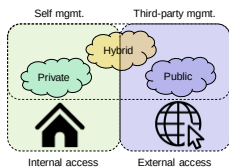
30

ifi

30

Deployment Models and Security

- A Cloud can be deployed in 3 ways (as discussed)
 - Public: publically accessible, pay-as-you-go
 - Private: internal resources in a single organization, better utilization of HW
 - Hybrid: private cloud → flash crowds/peaks → public cloud



Type	Infrastructure Mgmt.	Property	Localization	Security
Private	Self	Self	Internal	High
Hybrid	Self or third party	Self or third party	External or Internal	Medium
Public	Third party	Third Party	External	Low

© 2024 UZH, CSG@#

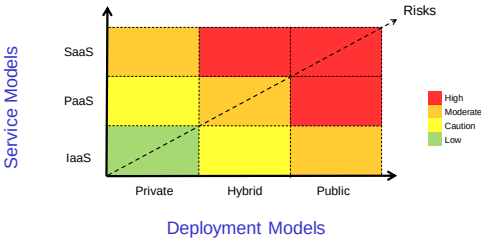
31



31

Security Responsibilities

- Security risks considering choice of service and deployment models



© 2024 UZH, CSG@#

32



32



OVERVIEW ON MAJOR CLOUD SECURITY THREATS

© 2024 UZH, CSG@#

33



33

Top Threats by Cloud Security Alliance

- CSA is non-profit organization to strengthen cloud computing security following an information security conference
 - Composed by industrial representatives, corporations, associations such as Dell, HP, and eBay
- Famous reports on threats based on a survey of its members:
 - First version in 2013 has 9 threats
 - "The Notorious Nine"
 - Latest version in 2020 has 11 threats
 - "Top Threats to Cloud Computing: The Egregious 11"



© 2024 UZH, CSG@#

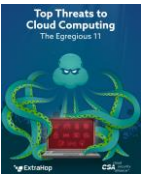
34



34

Overview of Threats

- Top Threats to Cloud Computing The Egregious 11
 - Data Breaches
 - Misconfiguration and Inadequate Change Control
 - Lack of Cloud Security Architecture and Strategy
 - Insufficient Identity, Credential, Access and Key Management
 - Account Hijacking
 - Insider Threat
 - Insecure Interfaces and APIs
 - Weak Control Plane
 - Metastructure and Applistructure Failures
 - Limited Cloud Usage Visibility
 - Abuse and Nefarious Use of Cloud Services



© 2024 UZH, CSG@#

35



35

Data Breaches



- Issue Definition:
 - A cybersecurity incident where sensitive, protected or confidential information is released, viewed, stolen or used by an unauthorized individual.
- Key Takeaways:
 - Data are becoming the main target of cyber attacks. Defining the business value of data and the impact of its loss is important for organizations that own or process data
 - Protecting data are evolving into a question of who has access to it.
 - Data accessible via the internet are the most vulnerable asset to misconfiguration or exploitation
 - Encryption techniques can help protect data, but negatively impact system performance while making applications less user-friendly
 - A robust and well-tested incident response plan that considers the CSP and data privacy laws will help data breach victims recover.



© 2024 UZH, CSG@#

36



36

Insufficient Identity, Credential, Access and Key Management



- Issue Definition:
 - Include tools and policies that allow organizations to manage, monitor and secure access to valuable resources. Incidents occur due to the following reasons:
 - Inadequate protection of credentials
 - Lack of regular automated rotation of cryptographic keys, passwords and certificates
 - Lack of scalable identity, credential and access management systems
 - Failure to use multifactor authentication
 - Failure to use strong passwords
- Key Takeaways:
 - Secure accounts, inclusive to two-factor authentication and limit the use of root accounts
 - Practice the strictest identity and access controls for cloud users and identities.
 - Segregate and segment accounts, virtual private clouds (VPCs) and identity groups based on business needs and the principle of least privilege
 - Rotate keys, remove unused credentials or access privileges and employ central, programmatic key management.

© 2024 UZH, CSG@#

39

ifi

39

Insecure Interfaces and APIs



- Issue Definition:
 - Cloud computing providers expose a set of software user interfaces (UIs) and APIs to allow customers to manage and interact with cloud services. The security and availability of general cloud services are dependent on the security of these APIs.
- Key Takeaways:
 - Practice good API hygiene. Good practice includes diligent oversight of items such as inventory, testing, auditing and abnormal activity protections.
 - Ensure proper protection of API keys and avoid reuse.
 - Consider using standard and open API frameworks (e.g., Open Cloud Computing Interface (OCCI) and Cloud Infrastructure Management Interface (CIMI))

© 2024 UZH, CSG@#

42

ifi

42

Cloud Security Guidelines

- Identify [main](#) security issues
- Establish priorities, norms and prevention measures
 - NIST: *Guideline on Security and Privacy in Public Cloud Computing*
 - CSA: *Security Guidance for Critical Areas of Focus in Cloud Computing v3.0*
 - ENISA: *Benefits, Risks and Recommendations for Information Security*
 - ISO 27001: *International security management standard – accreditation based on audits*



© 2024 UZH, CSG@#

47

ifi

47

CLOUD SECURITY GUIDELINE(S) AND TAXONOMY



© 2024 UZH, CSG@#

48

ifi

48

Virtualization Guidelines

- Provide a [general overview of virtualization security](#)
 - Identify major vulnerabilities and threats
 - Establish standards, priorities, and measures
- Guidelines:
 - NIST: *Guide to Security for Full Virtualization Technologies*
 - PCI Security Standards Council: *Data Security Standards (DSS) Virtualization Guideline*
 - Red Hat: *Virtualization Security Guide*



© 2024 UZH, CSG@#

49

ifi

49

Cloud Security Taxonomy

- Based on these guidelines, it is possible to summarize major aspects into 3 categories corresponding to the fundamental security principles
 - Architecture: security of all IT components
 - Compliance: legal obligations/procedures
 - Privacy: measures to safely handle/manage data



© 2024 UZH, CSG@#

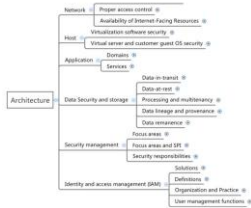
50

ifi

50

Architecture

- Security by design, involving all related IT components
- Requires a division of responsibilities between Providers and Customers:



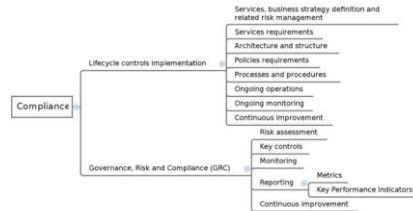
© 2024 UZH, CSG@#

51

ifi

Compliance

- Introduces administrative and legal responsibilities of the Provider toward offering cloud services



© 2024 UZH, CSG@#

52

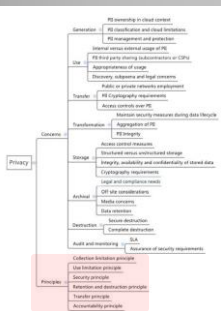
ifi

Privacy

- Divided into concerns and principles

– Concerns are related to the main issues in the lifecycle of data (e.g., creation, storage, disposal)

– Principles are related to best practices to ensure data privacy



© 2024 UZH, CSG@#

53

ifi

SUMMARY AND TAKEAWAYS



© 2024 UZH, CSG@#

54

ifi

Recommendations (1)

- A "Cloudification" needs to prepare and consider
 1. Strategic decisions
 - All or none, or a combination of in-house IT services plus Cloud services
 2. Qualitative effects
 - Clients' Quality-of-Experience (perceived Quality-of-Service)
 3. Quantitative impacts
 - In case of mission-critical applications, processing personal information, or handling otherwise sensitive information
 - Obtain legal advice wrt. to SLAs concluded
 - Investigate the cloud management and accounting system
 - Investigate preservation of the data in the Cloud
 - Run a risk assessment and affiliation-specific management analysis on top of those technical solutions available or planned for

© 2024 UZH, CSG@#

55

ifi

Recommendations (2)

- A more specific process includes check points
 - Identify assets for the Cloud deployment
 - (1) Data or (2) applications, functions, processes
 - Data and applications do not need to reside in the same location!
 - Evaluate each asset
 - Rough assessment of the value and sensitivity of each asset
 - Prepare the risk assessment and costs incurred planning
 - Potential uses of assets, determine threats/vulnerabilities
 - Map asset and uses to potential Cloud deployment scenarios
 - Public, private-internal, private-external
 - Evaluate concrete Cloud service models and providers
 - E.g., degree of control needed in IaaS, PaaS, SaaS, or FaaS
- Trade-off analysis

© 2024 UZH, CSG@#

56

ifi

Based on: H. M. Galliker: Startup in the Cloud, BA, February 2010

Cloud Characteristics

- ❑ Clouds do not define a single technology
- ❑ Clouds did not revolutionize technology
- ❑ Clouds determine a **business evolution** to match agile and flexible computing demands
- ❑ Clouds enable an IT “**capacity**” **increases** (e.g., storage, computing) while keeping (sometimes) costs low (or similar) without investing into new, e.g., infrastructure, maintenance, or software
- ❑ Clouds offer **multiple technological facets** and solutions

Potential
misconceptions

Reality

© 2024 UZH, CSG@UZH

57



57

Generic Cloud Evaluation

- + Scalability and encapsulated update/change management
 - Greater efficiency of provision as economies of scale hold
- + Costs
 - Pay-as-you-go scheme, which avoids over-buying resources
- + Flexibility and agility as well as next generation migration
 - On-demand computing demands
 - Ability to create new compute resources to experiment with
- **Security (Integrity, privacy, and data protection)**
 - Dedicated scenario- and technology-dependent case analysis required
- Availability, reliability, and lack of control
 - Full dependency on cloud provider (technology, SLA)
- Lock-in effects
 - Cloud provider-specific interfaces, maintenance, SLAs, and pricing exist

Mostly
positive aspects

Partially
negative aspects

© 2024 UZH, CSG@UZH

58



58

Computer Network and Security Principles (CNSP)

L05 – Access Control, Firewalls, Policies

© 2024 UZH, CSG@#

1

ifi

1

Access Control, Firewalls, Policies

- Objectives
 - To understand current concerns of access control
 - To get hold of the concept of firewalls and their types
 - To remember the key differences of security policies
 - Be able to express network-level security policies
- Topics
 - Major categorizations
 - Access control basics, characteristics, types
 - Firewall definitions, types, and deployment
 - Overview of network policies and their relation to security mechanisms (access control, firewall, IDS/IPS)

© 2024 UZH, CSG@#

2

ifi

2

Definition

- A "language" for expressing access control policies: who can access what, how and when ...
- Enforcement of access control
 - Identify all resources (objects) and their granularity
 - Identify all potential users (subjects)
 - Specify rules for subject/object interaction
 - Guard them in real time
- Focus
 - What a subject (user or process) is allowed to do
 - What may be done with an object (files, memory, printers, routers, disks, processes, and others)

© 2024 UZH, CSG@#

4

ifi

4

ACCESS CONTROL

© 2024 UZH, CSG@#

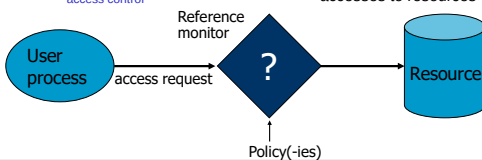
3

ifi

3

Characteristics

- Was one of the major security concerns in 70's and is still popular as of today:
 - Regulates information access
 - Governed by policies
 - Requires authentication
 - Authentication is a *subset* of access control
- Reference monitor mediate access to resources
 - Notion of complete mediation: control *all* accesses to resources



© 2024 UZH, CSG@#

5

ifi

5

Basic Concepts

- Conceptually, it is consider the premise that *users* accesses *objects*
- *Subject*: active entity that requests access to resources
 - User or process on behalf of the user
- *Object*: passive entity that store system information
 - Files, directories, memory segments
- *Access control*: mediation of subject requests to access objects
- *Reference monitor*: entity that receives subject requests and allow/deny access according to a security policy
 - Realizes the access control governed by a policy

© 2024 UZH, CSG@#

6

ifi

6

Types of Access Control (1)

- **Discretionary Access Control (DAC):**
 - Protect the information recipient but not the information itself.
 - Subjects can read information and freely disseminate independently.
 - Typically adopted in general purpose operating systems.
- **Mandatory Access Control (MAC):**
 - Typically protect the information via security policies.
 - Users cannot change the access control of information/resources
 - Data is characterized according to the level of confidentiality
- **Role-based Access Control (RBAC):**
 - Access rights are given based on roles and subjects receive permissions based on their roles in the system.
 - Widely used in security applications and databases.
- **Attribute-based Access Control (ABAC):**
 - According to NIST, ABAC is defined as an "Access control method where the subject requests for performing an operation on objects are granted/denied based on:
 - Assigned attributes of the subject.
 - Assigned attribute of the object.
 - Environmental conditions.
 - Set of policies.
 - ABAC uses attributes as the building blocks to define access control rules and access requests.
- **Rule-based Access Control:**
 - Not to be confused with RBAC.
 - Considers a dimension (spatial, temporal) to allow/deny access
 - Certain resources are accessible only during working time (e.g. 9am-6pm)
 - A rule is equivalent to a policy. DAC, MAC, RBAC are also governed by policies or rules.

© 2024 UZH, CSG@#

7

ifi

7

Types of Access Control (2)

Factors	DAC	MAC	RBAC	ABAC
Access Control to Information	Through owner of data	Through fixed rules	Through roles	Through attributes
Access Control Based on	Discretion of owner of data	Classification of users and data	Classification of roles	Evaluation of attributes
Flexibility for Accessing Information	High	Low	High	Very high
Access Revocation Complexity	Very complex	Very easy	Very easy	Very easy
Support for Multilevel Database System	No	Yes	Yes	Yes
Used in	Initial Unix system	The U.S. department of defense	ATLAS experiment in CERN	The Federal government

Khalid, Ernst F., and Mustafa M. Kadi. "A Survey of Access Control and Data Encryption for Database Security." *Journal of King Abdulaziz University* 28, no. 1 (2017): 19-30.

© 2024 UZH, CSG@#

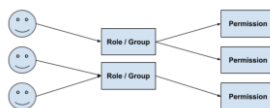
8

ifi

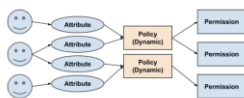
8

ABAC - an Evolution of RBAC

- **RBAC was proposed by NIST in 1992**
 - Ideal for small/medium enterprises
 - RBAC tends to not scale as people and resources are added, more roles are created to define more detailed permissions.



- **ABAC released in 2014 (NIST Guide to ABAC)**
 - Essentially, ABAC has a greater number of possible control variables than RBAC.
 - However, it is also more complex to be deployed (configured) and operated (processing power and time)



© 2024 UZH, CSG@#

9

Guide to Attribute Based Access Control (ABAC) Definition and Considerations, SP 800-162. <https://nvd.nist.gov/topics/sp800-162/publications/nist-sp-800-162.pdf>

ifi

9

Access Control Matrix (ACM) Model

- **Conceptual model underlying the DAC**
 - Describes protection state precisely
 - Rights/permissions of subjects (i.e., processes and users)
 - State transitions change elements of matrix
- **System protection state is represented by an access matrix**

Subjects	Objects (entities)					
	O ₁	...	...	...	...	O _n
S ₁						
...						
S _n						

- Subjects $S = \{s_1, \dots, s_n\}$
- Objects $O = \{o_1, \dots, o_m\}$
- Rights $R = \{r_1, \dots, r_k\}$
- Entries $A[s_i, o_j] \subseteq R$
- $A[s_i, o_j] = \{r_{x_1}, \dots, r_{x_y}\}$ means subject s_i has rights $r_{x_1}, \dots, r_{x_y}$ over object o_j

© 2024 UZH, CSG@#

10

ifi

10

Example of ACM Permissions

- **Unix**
 - r = read
 - w = write
 - x = execute
- **AFS (Andrew File System)**
 - r = read
 - l = list
 - i = insert
 - d = delete
 - w = write
 - k = lock
 - a = administer
- **Example:**
 - Processes
 - p, q
 - Files
 - f, g
 - Rights
 - $r, w, x, a(ppend), o(wn)$

Subjects	Objects (entities)			
	f	g	p	q
p	rwo	r	rwXO	w
q	a	ro	r	rwXO

© 2024 UZH, CSG@#

11

ifi

11

Implementation of ACM

- **ACMs are typically sparse. Thus, rarely implemented using matrices**
 - Suppose we have 1k 'users' and 100k 'files' and users should only read/write their own files
 - ACM will have 101K columns and 1k rows
 - Most elements would be empty or identical
- **Three practical approaches:**
 - Access Control Lists (ACLs)
 - List of capabilities
 - Lock and key

© 2024 UZH, CSG@#

12

ifi

12

Access Control Lists (ACLs)

- Most common implementation
- Each object is associated with an ACL that indicates for each subject in the system the permissions over objects
 - Corresponds to store an ACM by its columns



Subjects	Objects (entities)				
	f	g	Dsds	sdsd	
p	rwo	r	rwxo	w	
q	a	ro	r	rwxo	

ACM

f: (p, {rwo}), (q, {a})
g: (p, {r}), (q, {a})
p: (p, {rwxo}), (q, {r})
q: (p, {w}), (q, {rwxo})

ACL

List of Capabilities

- Each subject is associated to a list (of capabilities) that indicates access permissions to objects the subject is authorized
 - Corresponds to store an ACM by its lines



Subjects	Objects (entities)				
	f	g	p	q	
p	rwo	r	rwxo	w	
q	a	ro	r	rwxo	

ACM

p: (f, {rwo}), (g, {r}), (p, {rwxo}), (q, {r})
q: (f, {a}), (g, {ro}), (p, {r}), (q, {rwxo})

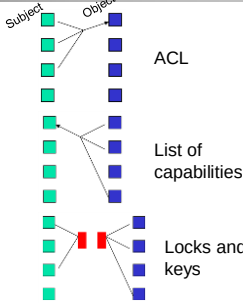
List of capabilities

ACLs and Capabilities

- Theoretically equivalent; consider 2 questions:
 - Given a subject, what objects can it access, and how?
 - Given an object, what subjects can access it, and how?
- ACLs answers the second, list of capabilities, the first
 - As the first questions gains relevance (e.g., incident response), this may change
- ACLs facilitates the review and revocation of permissions associated with objects
 - Revocation of subject's permissions require the inspection of all ACLs
- Lists of capabilities facilitates the revocation of permissions associated with subjects
 - Revocation of object's permissions require the inspection of all capabilities

Locks and Keys

- Associate lock with object and key with subject
 - Lock controls what the subject can access and how
 - Subject presents key; if it corresponds to any of the locks on the object, access is granted
- Hybrid between ACL and Capabilities
 - Every subject has a capability list that contains tuples of the form (o, key)
 - Object "o" accessed with key
 - Every has an access control list that contains tuples of the form (lock, mode)
 - Lock can be accessed for a given mode (r,w,x)



ACL, Capabilities, Lock-key

Feature/Method	Access Control List (ACL)	List of Capabilities	Lock-Key
Concept	List of objects detailing which subjects can access it and how.	List of subjects detailing which objects it can access and how.	An object (lock) can only be accessed if the correct key is present.
Delegation	Requires modifying the object's ACL.	Easier, just pass the capability to another subject.	Key can be transferred or copied to another subject.
Revocation	Relatively simple: remove the subject from the object's ACL.	More complex; must ensure capability is removed from all subjects who have it.	Relatively simple, change the lock or invalidate the key.
Typical Use	File systems, databases	Operating systems, microservices	Physical access control, some digital authentication systems.
Strengths	Centralized control, clarity on who has access to an object.	Flexibility in granting access rights, efficient for systems where subjects frequently change permissions.	Simplicity, clear delineation of access.
Weaknesses	Scalability issues in systems with numerous objects and subjects, delegation can be complex.	Difficulty in revoking access rights once granted, potential for "capability leak".	Limited granularity, revocation means changing locks or invalidating keys.

Summary

- "Access control" is where security engineering meets computer science
 - Control which principals (persons, processes, machines) have access to which resources in the system
- Access control rights can be specified in the form of
 - Access control matrix
 - Access control lists
 - Capability lists
 - Access control keys and locks
- Types of access control: DAC, MAC, RBAC, ABAC
- Common issues
 - Little protection when attacker has physical access
 - Poorly configured policies can be a problem
 - No defense against stack-based attacks

FIREWALL

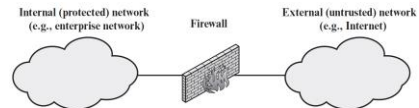
© 2024 UZH, CSG@#

19

ifi

Definition

- An integrated collection of security measures designed to control traffic flow into and out of a network
 - Similar to firewalls in building construction
 - Both are intended to isolate one "network" or "compartment" from another
- Firewall protects from general probes and many attacks



© 2024 UZH, CSG@#

20

ifi

The Need for Firewalls

- Majority of hosts have vulnerabilities
 - Software bugs, zero-day vulnerabilities, reduce exposure to unknown threats
 - Firewalls isolate possible threats
- Firewalls *execute less code*, and thus, are less prone to vulnerabilities
 - Managed by specialized security staff
 - Implement a logical separation of a network in different organizational domains
 - UZH → OEC → ifi → CSG, each one can deploy a separate firewall
 - Only few (and protected) machines are exposed to external traffic
- Design goals:
 - All traffic from inside to outside must pass through the firewall
 - Only authorized traffic defined by security policies will be allowed to pass
 - The firewall itself is immune to penetration



Capabilities:

- Defines a single choke-point
- Provides a location for monitoring security events
- Convenient platform for several Internet functions that are not security related

Limitations:

- Cannot protect against advanced attacks
- Cannot protect against internal threats
- Improperly secured WLAN can be accessed externally
- Internet hotspots may also breach firewalls

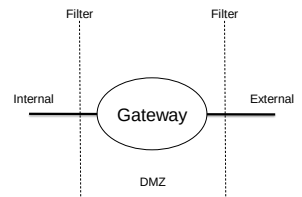
© 2024 UZH, CSG@#

21

ifi

Scenario

- Internal
 - Trusted premises
- External
 - Untrusted premises
- DMZ (DeMilitarized Zone)
 - Logical address space where needed but potentially danger servers are exposed
 - Example: Web, DNS, and e-mail servers



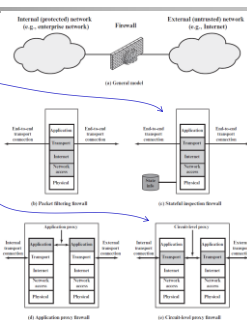
© 2024 UZH, CSG@#

22

ifi

Firewall Types

- Packet filtering
- Stateful inspection
- Application proxy
- Circuit-level



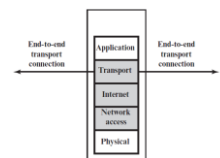
© 2024 UZH, CSG@#

23

ifi

Packet Filtering Firewalls

- Applies rules to each incoming and outgoing IP packet
 - Typically a list of rules based on matches in the IP or TCP header
 - Forwards or discards the packet based on rules match
- Two default policies
 - **Discard:** prohibit unless expressly permitted
 - More controlled, visible to users
 - **Forward:** permit unless expressly prohibited
 - Easier to manage and use but less secure



(b) Packet filtering firewall

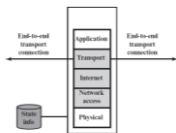
Rule	Direction	Source Address	Destination Address	Protocol	Destination Port	Action
A	In	External	Internal	TCP	23	Permit
B	Out	Internal	External	TCP	> 1023	Permit
C	Out	Internal	External	TCP	23	Permit
D	In	External	Internal	TCP	> 1023	Permit
E	Either	Any	Any	Any	Any	Deny

© 2024 UZH, CSG@#

24

ifi

Stateful Inspection Firewalls

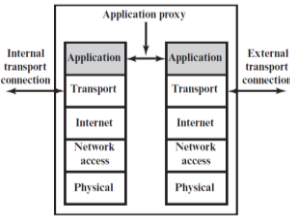


(c) Stateful inspection firewall

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.22.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.106	1033	173.166.32.222	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	80	Established
31.32.32.123.32	2112	192.168.1.6	80	Established
209.92.212.10	321	192.168.1.6	80	Established
34.102.32.23	1025	192.168.1.6	80	Established
223.21.22.12	1046	192.168.1.6	80	Established

- Goal:
 - Determine when packets are part of legitimate sessions originating from a trusted network
- Maintain tables containing active connections
 - IP addresses
 - Ports
 - Sequence numbers
- Using these tables, stateful firewalls can allow only inbound TCP packets that are originated in response to the internal network initiated connections

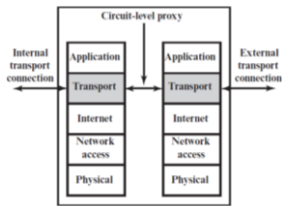
Application-proxy Firewall



(d) Application proxy firewall

- Also termed application-level
- Act as a relay of application traffic
 - User contacts gateway using a TCP/IP application
 - User is authenticated
 - Gateway contacts application on remote host and relays TCP segments between server and user
- Must have proxy code for each application
 - May restrict application features
 - Tend to be more secure than packet filters
- Disadvantage
 - More complex to be implemented
 - Additional processing overhead on each connection

Circuit-level Firewalls



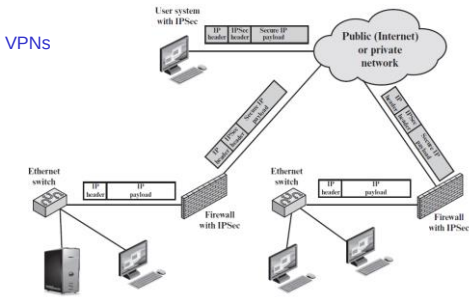
(e) Circuit-level proxy firewall

- Approach:
 - Set up two TCP connections, one between itself and a TCP user on an inner host and one on an outside host
 - Relays TCP segments from one connection to the other without examining contents
 - Security function consists of determining which connections will be allowed
- Typically used when inside users are trusted
 - May use application-level gateway inbound and circuit-level gateway outbound
 - Lower overheads

Firewall Topologies

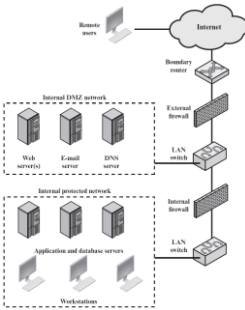
Host-resident firewall	Includes personal firewall software and firewall software on servers
Screening router	Single router between internal and external networks with stateless or full packet filtering
Single bastion inline	Single firewall device between an internal and external router
Single bastion T	Has a third network interface on a bastion to a DMZ where externally visible servers are placed
Double bastion inline	DMZ is sandwiched between bastion firewalls
Double bastion T	DMZ is on a separate network interface on the bastion firewall
Distributed firewall configuration	Used by large business and governments organizations

Single Bastion Inline

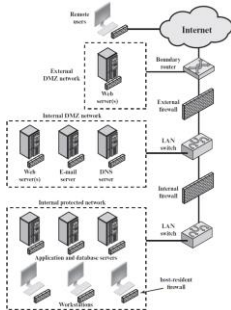


Double Bastion Inline

- DMZ Networks
 - Common deployment for enterprise networks
 - Single DMZ zone hosting externally accessible services
 - Internal firewall protects internal apps from possible malware and threats at the DMZ and the DMZ from internal threats



Distributed Firewalls



- Involves stand-alone firewall plus host-based firewalls working together under a central administrative control
- Typically involves an internal and external DMZ
 - Most external DMZs typically hosting Web servers

Summary

- Firewalls
 - Critical service to isolate internal networks and services
 - Types of firewalls, deployment
 - DMZ Networks
- Firewall blocks unauthorized network access
 - Not a (efficient) standalone solution, better when combined with additional mechanisms such as IDS
- Only effective if correctly configured with a precise definition of policies
 - May use several configurations to protect and isolate networks

Definition

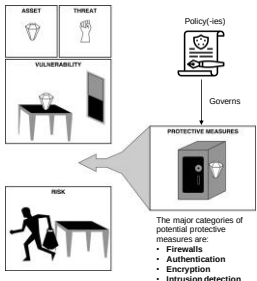
- A policy is a rule, or a set of rules that are used to manage and control the changing and/or maintaining of the state of one or more managed objects
- Policy-based management (PBM) is a management paradigm that separates the rules that govern the behavior of a system from the functionality of the system
 - PBNM allows for:
 - Automated decision processes
 - Network management simplification
 - Able to handle security issues



SECURITY POLICIES

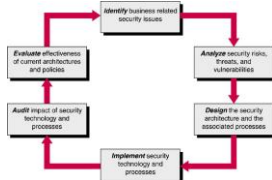
Network Security

- A successful network security implementation requires a marriage of technology and process.
 - "Roles and responsibilities and corporate standards for business processes and acceptable network-related behavior must be clearly defined, effectively shared, universally understood, and vigorously enforced for implemented network security technology to be effective."
- Process definition and setting of corporate security standards must precede technology evaluation and implementation.

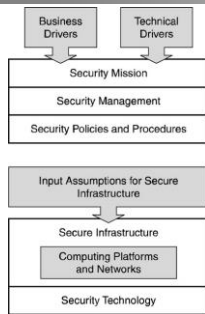


Security Policies

- System perspective:
 - Policies define appropriate behavior (and also inadequate system behavior)
 - Policies set the stage in terms of what tools and procedures are needed and allowed to be used by expressed set of subjects.
- Governance perspective:
 - Policies communicate a consensus over a desired security state
 - Policies provide a foundation for HR action in response to inappropriate behavior.
 - Policies may help prosecute cases



Security Architecture



- A representative example of a security architecture that clearly maps business and technical drivers through security policy and processes to implemented security technology.

© 2024 UZH, CSG@#

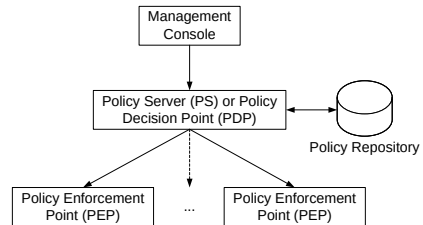
37

ifi

37

Policy-based Network Management

- General architecture for a PBNM system was described by IETF (Internet Engineering Task Force)



© 2024 UZH, CSG@#

38

ifi

38

Policy Abstraction

- Policies can be represented in different levels of abstraction

Value of Network Security Policy Development	Explanation/Implication
Identification/Authentication	Want to be assured that users can be accurately identified and that only authenticated users are allowed access to corporate resources
Access Control/Authentication	Want to be assured that even authenticated users are only allowed access to those information and network resources that they are supposed to access
Privacy/Confidentiality	Want to be assured that network based communication is private and not subject to eavesdropping
Data Integrity	Want to be assured that data is genuine and cannot be changed without proper controls
Nonrepudiation	Want to be assured that users cannot deny the occurrence of given events or transactions

How to translate high-level goals into network actions?

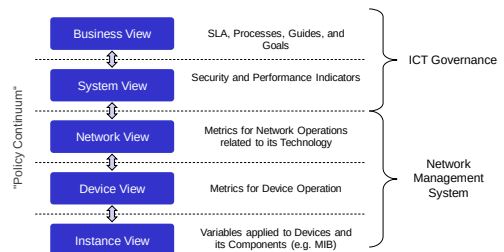
© 2024 UZH, CSG@#

39

ifi

39

Policy Continuum



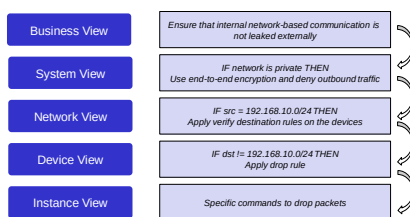
© 2024 UZH, CSG@#

40

ifi

40

Firewall Example



© 2024 UZH, CSG@#

41

ifi

41

Towards Autonomic Security Management

- Recent networking technologies such as [Software-Defined Networks \(SDN\)](#), [Virtual Networking Function \(VNFs\)](#), and the [DevOPS tooling](#) have been extensively used to enable autonomic networks
- High-level policies are expressed as [intents](#)
 - Intent-based Network (IBN) is a widely explored concept
- SDN can «easily» (in contrast to traditional, configuration-based management systems) embrace security policies and define secure states that a system will try to achieve
 - VNFs can be defined as isolated functions (firewall, IDS) and run as containers orchestrated by SDN
- Nonetheless, [complexity to achieve an autonomous system is extremely high](#)
 - Partial automation assisted by operators is factible

© 2024 UZH, CSG@#

42

ifi

42

Summary

- Policies express desired system states (not only security-related)
 - Major interface transform business goals into network actions
- Policies do not solve any new system vulnerability and may introduce new vulnerabilities if mistakenly defined
 - Express wrong policies may allow/deny unintended traffic
 - Still, it would not solve a particular application vulnerability
- There are well-defined concepts, architectures, and tools to support autonomic management
 - PBNM, IBN, SDN, NFV, Docker, ML/AI, and many more acronyms!
 - ... At the price of increased software complexity

© 2024 UZH, CSG@ifi

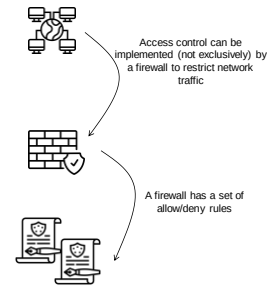
43



43

Takeaways

- Access Control
 - Mechanism or model that restricts access to information, services in a system
- Firewall
 - Mechanism that restricts access of traffic based on a set of allow/deny rules
- Policy(-ies)
 - Set of rules defining system behavior



© 2024 UZH, CSG@ifi

44



44

Computer Network and Security Principles (CNSP)

L06 – Intrusion Detection Systems

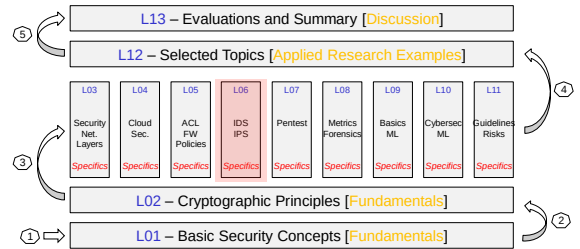
© 2024 UZH, CSG@#

1

ifi

1

Overview of Class and Lectures (Chapter Dependency)



© 2024 UZH, CSG@#

2

ifi

2

Intrusion Detection Systems (IDS)

- Objectives
 - To understand current concerns of IT systems' usage
 - To get hold of the concept of IDS/IPS
 - To remember the key differences of IDS categorization
 - Based on signature information or
 - Anomaly generating false detections
 - To provide an overview of the Snort IDS
- Topics
 - Major categorizations
 - IDS basics and operation
 - Applied example based on Snort
 - Summary of recommendations and takeaways

© 2024 UZH, CSG@#

3

ifi

3

MOTIVATION

© 2024 UZH, CSG@#

4

ifi

4

Today's Concerns

- Internet-based secure information transmission key
 - E.g., and especially for healthy reputation and financial status
 - Risks and chances of malicious interactions increase
- Intrusion attempt as possibility of a deliberate unauthorized attempt to:
 - Access information
 - Manipulate information
 - Render a system unreliable or unusable
- Countermeasure
 - To detect attacks asap and to take appropriate action
 - IDS does not take preventive measures, but it is rather reactive rather than pro-active, "informant" not "policing"

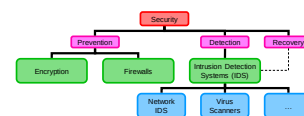
© 2024 UZH, CSG@#

5

ifi

5

Where Intrusion Detection Fit into the Security Landscape?



- Principle: Defense in Depth (layered mechanisms)



© 2024 UZH, CSG@#

6

ifi

6

Current Situation

- **Most popular** way to detect intrusions via **audit data** generated by the operating system
 - Audit trail is a record of activities on a system that are logged to a file in chronologically sorted by order (time)
 - Audit trails useful in establishing guilt attackers
 - Audit trails are often the only way to detect unauthorized, but subversive user activity.
- **Selected statements**
 - Information theft (i.e., data leak) is up over 250% in the last 5 years
 - 99% of major companies report at least one major incident
 - Telecom, computer fraud totaled \$10 billion in the US alone
- Subversion cannot be prevented, **detection** is key

Eugene Spafford: <https://spaf.cerias.purdue.edu/quotes.html>

© 2024 UZH, CSG@#

7

ifi

7

IDS FUNDAMENTALS

© 2024 UZH, CSG@#

8

ifi

8

Definition

- **Intrusion**
 - A set of actions aimed to compromise security goals, namely integrity, confidentiality, or availability, of a computing and networking resource
- **Intrusion Detection System (IDS)**
 - The process of identifying and responding to intrusion activities
 - Reactive set of measures
- **Intrusion Prevention System (IPS)**
 - Extension of ID with exercises of access control to protect computers from exploitation
 - Proactive set of measures

© 2024 UZH, CSG@#

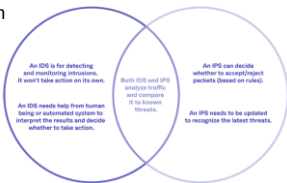
9

ifi

9

Example

- **IDS** is what you do after prevention has failed
 - Detect attack in progress
 - Network traffic patterns, suspicious system calls, etc.
 - Discover telltale system modifications
- **IPS**
 - Find buffer overflows and remove them
 - Use firewall to filter out malicious network traffic



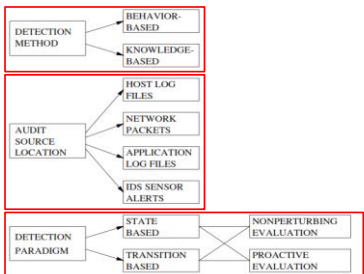
© 2024 UZH, CSG@#

10

ifi

10

IDS Taxonomy



Based on: H. Debar, An Introduction to Intrusion-Detection System, IBM Research Zurich

© 2024 UZH, CSG@#

11

ifi

11

Major Assumptions

- **Primary assumptions**
 - System activities are **observable**
 - Measurable parameters determined and defined
 - Monitoring components exist
 - Normal and intrusive activities show **distinct** evidence
 - Determined "in advance"



• Can you think in one example where normal and intrusive activities are easily identified?

© 2024 UZH, CSG@#

12

ifi

12

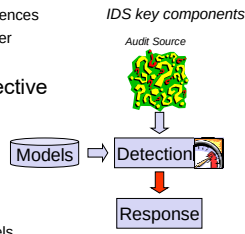
Components

Components of intrusion detection systems

- From an algorithmic perspective
 - Features – capture intrusion evidences
 - Models – piece evidences together

System's architecture perspective

- Components required
 - Audit data processor
 - Knowledge base
 - Decision engine
 - Alarm generation
 - Responses and feedback channels



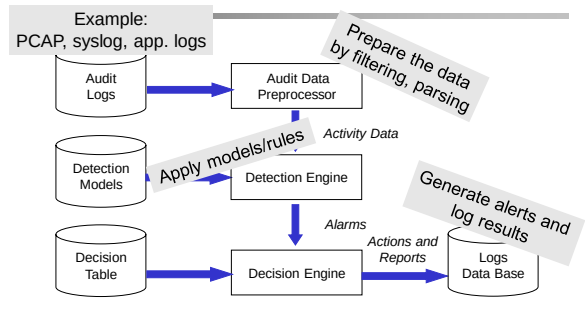
© 2024 UZH, CSG@#

13

ifi

13

Components' Interactions



Based on: H. Debar, An Introduction to Intrusion-Detection System, IBM Research Zürich

© 2024 UZH, CSG@#

14

ifi

14

Intrusion Detection Approaches and Monitoring

Modeling

- Features
 - Evidence extracted from audit logs
- Analysis approach
 - Piecing evidence together
 - Misuse detection (signature-based)
 - Anomaly detection (statistics-based)

Deployment

- Network-based
 - Monitor network traffic between nodes
- Host-based
 - Monitor nodes' processes

Monitoring options:

- Operating System calls
- Command lines
- Network data
 - E.g., from routers and firewalls
- Processes
- Keystrokes
- File and device accesses
- Memory accesses
- Key concern
 - All monitoring (and auditing) has to be **scalable!**
 - Run-time monitoring with replication and non-determinisms

© 2024 UZH, CSG@#

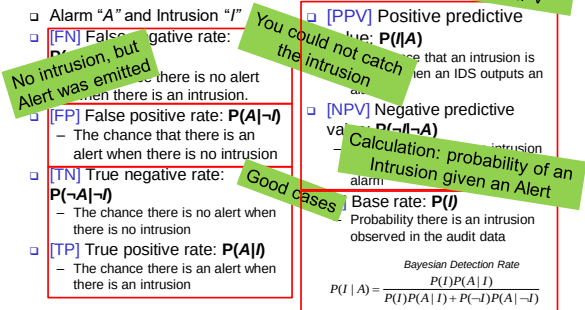
15

Detection of malicious behavior via System Call Monitoring
<https://www.cis.drexel.edu/~spencer/papers/sp2013.pdf>

ifi

15

Example of IDS Metrics



© 2024 UZH, CSG@#

16

ifi

16

DETECTION TECHNIQUES, TYPES, AND CLASSIFICATIONS

© 2024 UZH, CSG@#

17

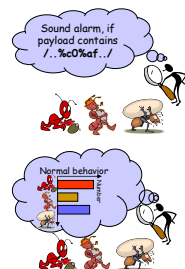
ifi

17

Detection Techniques

Different ways of classifying an IDS based on

- Misuse detection (signature-based)
 - Define what is **wrong** and give alarms for such behavior. (default permit)
- Anomaly Detection
 - Define what is **correct** and give alarms for everything else. (default deny)



© 2024 UZH, CSG@#

18

ifi

18

Passive and Active Detection

Passive systems

- To send alarm to administrator in case of a detected attempt

Active system

- To take control of the situation by **disconnecting** the assailant
- Methods:

- Session Disruption

- IDS may send a TCP reset packet if the attacker has opened a TCP connection to the victim
- IDS may send various UDP packets to disrupt a UDP connection
- Will not permanently remedy the situation (disconnect only)

- Rule Modification

- IDS is linked to a firewall via an administrative link
- IDS communicates with the firewall telling it to drop all packets from the attackers IP Address

© 2024 UZH, CSG@#

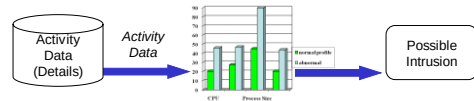
ifi

19

Anomaly Detection

Define a profile describing “normal” behavior

- Detecting deviations
- Potentially able to detect new attacks



(Relatively) high false positive rates

- Anomalies may be similar to “new” normal activities
- Anomalies may be caused by other element faults
 - E.g., a router failure, a P2P or misconfiguration, different users
- Which method will detect DDoS SYN flooding?

© 2024 UZH, CSG@#

20

ifi

20

Anomaly Detection Types

Static anomaly detection

- Assumes that code “sections” on host remain constant
- Focuses only on software side and ignores unusual changes in hardware
- Used to monitor data integrity

Dynamic anomaly detection

- Depends on a “baseline” or “profile”
 - The baseline is established by IDS or network administrator
- A baseline determines, which traffic is normal
- Possible information may include
 - E.g., Bandwidth, ports, time frames, or system calls

© 2024 UZH, CSG@#

21

ifi

21

Anomaly Detection IDS

Summary:

- This IDS **models** the normal usage of the network as a **noise characterization**.
- Anything distinct from the noise is assumed to be an intrusion activity.
 - E.g flooding a host with lots of packet.
- **The primary strength is its ability to recognize novel attacks.**

Drawbacks:

- Assumes that intrusions will be accompanied by manifestations that are sufficiently unusual so as to permit detection.
- These generate many false alarms and hence compromise the effectiveness of the IDS.

© 2024 UZH, CSG@#

22

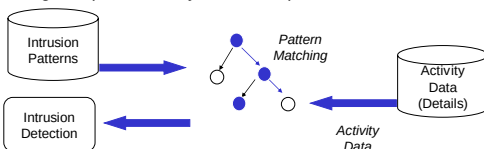
ifi

22

Signature-based

Intrusion patterns (attack signatures) include

- E.g., sequences of system calls, patterns of network traffic



Example:

- **if** (traffic contains “x80+de[“\r\n”]{20}”) **then** “attack detected”
- High accuracy on existing patterns, needs to know “in-advance”
- But cannot detect new attacks

© 2024 UZH, CSG@#

23

ifi

23

Extraction of Misuse Signatures

Using “invariant” characteristics of attacks known

- E.g., bodies of known viruses and worms, port numbers of applications with known buffer overflows, RET addresses of overflow exploits
- Hard to handle mutations?
 - Polymorphic viruses: each copy has a different body

Key research challenge

- Fast and automatic extraction of signatures of new attacks
 - Honeypots are useful
 - Attracting malicious activity
 - To become an early target

© 2024 UZH, CSG@#

24

ifi

24

Signature-based IDS

□ Summary:

- This IDS possess an attack description that can be matched to sensed attack manifestations.
- The question of what information is relevant to an IDS depends upon what it is trying to detect.
 - E.g DNS, FTP etc.

□ Drawbacks:

- They are unable to detect novel attacks.
- Suffer from false alarms
- Must be programmed again for every new pattern to be detected.

© 2024 UZH, CSG@#

25

ifi

25

IDS Types

□ Traditionally divided in two different ways:

- Based on the placement (implying input/audit data)
 - Host-based versus network-based IDS
- Based on the type of detection paradigm
 - Model *signature* versus *anomalous* behavior



□ Modern systems are a *mixture*

- Hybrid system using both normal and anomalous models, and
- Application-based, etc.

© 2024 UZH, CSG@#

26

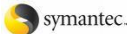
ifi

26

Host-based IDS (HIDS)

□ Operating System monitoring/analysis mechanisms and auditing to detect "misuse" in terms of code

- Executes *fully static and dynamic analysis of a program*
 - Monitoring shell commands and system calls executed by user applications and system programs
- *Most comprehensive* program information available for detection, thus, considered (highly) accurate



□ Drawbacks

- User-dependent: IDS required on all user machines
- Local view of an attack observed
 - Covering only a limited IP (Internet Protocol) address space: scalability
- Worst case: Attacker takes over machine
 - Tampering with IDS itself and possibly modifying audit logs



© 2024 UZH, CSG@#

27

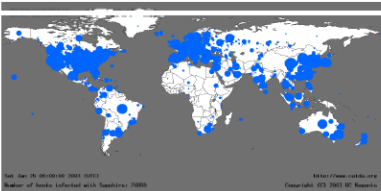
ifi

27

Example – SQL Slammer (1)

□ SQL Slammer, 2003

- Saturday, January 25, 2003: just before 5:30am (UTC), say 6:30am (MEZ) first instance of worm discovered
- Large-scale distribution from 6:00am (UTC) onwards



<https://www.caida.org/publications/papers/2003/sapphire/sapphire.html>

© 2024 UZH, CSG@#

28

ifi

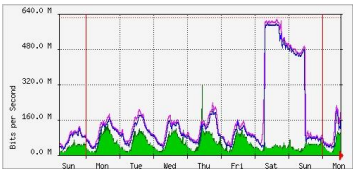
28

Example – SQL Slammer (2)

□ Munich Research Network (MWN)

- For local Munich educational units
- Massive problems from Saturday January 25, 2003, at 6.30am until Sunday January 26, 2003, at 11.30am

□ Traffic statistics at central backbone router (1 week)



- Output traffic
- Input traffic (maximum peak for every 5 min)
- Output traffic (maximum peak every 5 min)

© 2024 UZH, CSG@#

29

ifi

29

HIDS Summary

□ Summary:

- The host operating system or the application logs in the audit information.
 - These audit information includes events like the use of identification and authentication mechanisms (logins etc.), file opens and program executions, admin activities etc.
- This audit is then analyzed to detect trails of intrusion.

□ Drawbacks:

- The kind of information needed to be logged in is a matter of experience.
- *Unselective logging of messages may greatly increase the audit and analysis burdens.*
- Selective logging runs the risk that attack manifestations could be missed.

© 2024 UZH, CSG@#

30

ifi

30

Network-based IDS (NIDS)

- ❑ To deploy monitors at strategic locations
 - E.g., packet sniffing via `tcpdump` at routers
- ❑ To inspect network traffic, protecting multiple hosts
 - E.g., violations of protocols, unusual connection patterns
 - E.g., **Deep Packet Inspection (DPI)** of payload
- ❑ Limitations
 - Monitoring only within router or firewall
 - Cannot execute payload or perform code analysis
 - Even DPI gives limited application-level semantic information
 - To record and process vast amount of traffic data
 - Easily defeated by encrypting traffic
 - Possible mitigation with encryption only at gateway

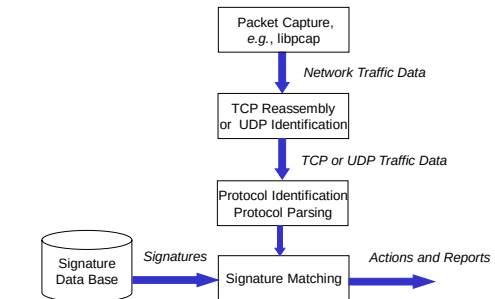


© 2024 UZH, CSG@#

31



NIDS Architecture



© 2024 UZH, CSG@#

32



31

32

NIDS Summary

- | | |
|--|--|
| <ul style="list-style-type: none"> ❑ Summary: <ul style="list-style-type: none"> – This IDS looks for attack signatures in network traffic via a promiscuous interface. – A filter is usually applied to determine which traffic will be discarded or passed on to an attack recognition module. <ul style="list-style-type: none"> • Helps to filter out known un-malicious traffic. | <ul style="list-style-type: none"> ❑ Drawbacks: <ul style="list-style-type: none"> – No insight on specific application or protocol behavior <ul style="list-style-type: none"> • May trigger false alarms – Difficult to operate on encrypted traffic <ul style="list-style-type: none"> • Possible to apply differential cryptanalysis in advanced, next-generation NIDS – Drawback between performance and accuracy |
|--|--|

© 2024 UZH, CSG@#

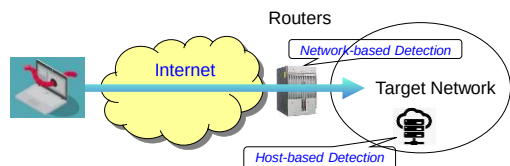
33



33

Host-based vs. Network-based IDS

- ❑ With "early" worm stages, only limited worm samples
 - Host-based IDS with limited IP address space
- ❑ Network-based might be able to detect worms in their "early" stage



© 2024 UZH, CSG@#

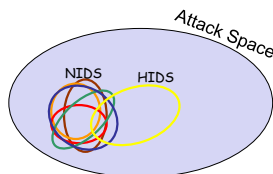
34



34

Hybrid Intrusion Detection

- ❑ Possibility to combine not only different types, but also detection techniques
 - **Detection**
 - Signature(misuse)-based
 - Anomaly-based
 - **Type**
 - Network-based (NIDS)
 - Host-based (HIDS)



© 2024 UZH, CSG@#

35



35

EXAMPLES OF OPEN SOURCE IDS TOOLS

© 2024 UZH, CSG@#

36



36

Open Source IDS Tools

- ❑ Snort
 - <https://www.snort.org/>
- ❑ Suricata
 - <https://suricata.io/>
- ❑ Zeek (formerly known as Bro)
 - <https://zeek.org/>
- ❑ OSSEC
 - <https://www.ossec.net/>
- ❑ Samhain Labs
 - <https://www.la-samhna.de/samhain/>



© 2024 UZH, CSG@ifi

37

ifi

39

Introduction to Snort

- ❑ Snort is a multi-mode packet analysis tool
 - Developed out of the evolving need to perform network traffic analysis in both *real-time* and for *forensic post processing*
- ❑ Three main operational modes
 - Sniffer Mode
 - Like tcpdump or wireshark
 - Packet Logger Mode
 - Forensic Data Analysis Mode
 - NIDS Mode
 - Mirror port to a host/server running Snort
- ❑ Snort characteristics:
 - Small (~800k source download)
 - Portable (Linux, Windows, MacOS X, Solaris, BSD, IRIX, Tru64, HP-UX, etc)
 - Fast (High probability of detection for a given attack on 100Mbps networks)
 - Configurable (Easy rules language, many reporting/logging options)
 - Free (GPL/Open Source Software)

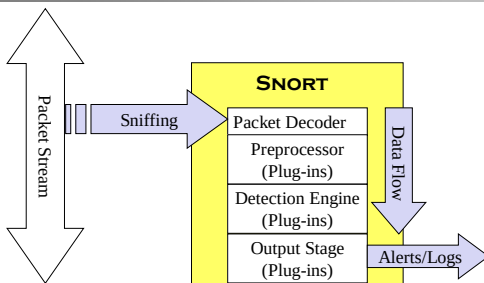
© 2024 UZH, CSG@ifi

38

ifi

38

Data Flow



© 2024 UZH, CSG@ifi

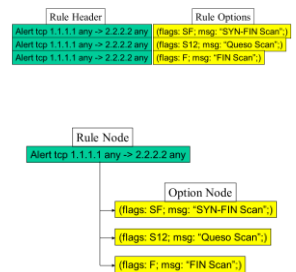
39

ifi

39

Detection Engine: Rules

- ❑ Implemented as a 3-dimensional linked list
 - Dimensions 1 & 2 contain data nodes to be tested against current packet
 - Dimension 3 contains linked lists of function pointers to test the node's data against the packet
- ❑ Entire engine is walked recursively
 - Very fast, very robust
 - "First exit" detection strategy
 - First detect causes engine to perform rule action & then go on to next packet



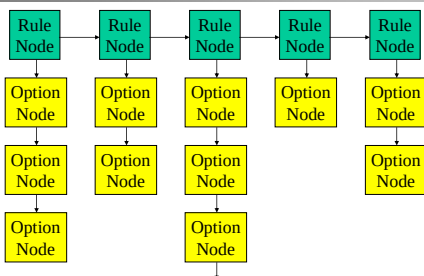
© 2024 UZH, CSG@ifi

40

ifi

40

Detection Engine: Fully Populated



© 2024 UZH, CSG@ifi

41

ifi

41

Snort Rules

- ❑ Snort rules are extremely flexible and are easy to modify, unlike many commercial NIDS
 - Sample rule to detect SubSeven trojan:

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR
subseven 22"; flags: A+; content: "|0d0a5b52504c5d3030320d0a|";
reference:arachnids,485; reference:url,www.hackfix.org/subseven/;
sid:103; classtype:misc-activity; rev:4;)
```

- ❑ Elements before parentheses comprise 'rule header'
- ❑ Elements in parentheses are 'rule options'

© 2024 UZH, CSG@ifi

42

ifi

42

Rules Header

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR
subseven 22"; flags: A+; content: "|0d0a5b52504c5d3030320d0a|";
reference:arachnids,485; reference:url,www.hackfix.org/subseven/;
sid:103; classtype:misc-activity; rev:4;)
```

- ❑ alert action to take; also log, pass, activate, dynamic
- ❑ tcp protocol; also udp, icmp, ip
- ❑ \$EXTERNAL_NET source address; this is a variable – specific IP is ok
- ❑ 27374 source port; also any, negation (!21), range (1:1024)
- ❑ -> direction; best not to change this, although <> is allowed
- ❑ \$HOME_NET destination address; this is also a variable here
- ❑ any destination port

Rules Options

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR
subseven 22"; flags: A+; content: "|0d0a5b52504c5d3030320d0a|";
reference:arachnids,485; reference:url,www.hackfix.org/subseven/;
sid:103; classtype:misc-activity; rev:4;)
```

- ❑ msg:"BACKDOOR subseven 22"; message to appear in logs
- ❑ flags: A+; tcp flags; many options, like SA, SA+, !R, SF*
- ❑ content: "|0d0..0a|"; binary data to check in packet; content without | (pipe) characters do simple content matches
- ❑ reference...; where to go to look for background on this rule
- ❑ sid:103; rule identifier
- ❑ classtype: misc-activity; rule type; many others
- ❑ rev:4; rule revision number
- ❑ other rule options possible, like offset, depth, nocase

Many Examples of Rules

- ❑ bad-traffic.rules exploit.rules scan.rules
- ❑ finger.rules ftp.rules telnet.rules
- ❑ smtp.rules rpc.rules rservices.rules
- ❑ dos.rules ddos.rules dns.rules
- ❑ tftp.rules web-cgi.rules web-coldfusion.rules
- ❑ web-frontpage.rules web-iis.rules web-misc.rules
- ❑ web-attacks.rules sql.rules x11.rules
- ❑ icmp.rules netbios.rules misc.rules
- ❑ backdoor.rules shellcode.rules policy.rules
- ❑ porn.rules info.rules icmp-info.rules
- ❑ virus.rules local.rules attack-responses.rules

DISCUSSION AND TAKEWAYS

Deploying and IDS

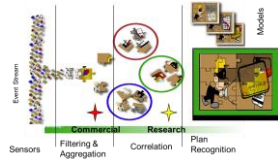
- ❑ IDS != FW
 - Not something you just drop into your network and then leave there
 - Extensive tuning needed
 - Some regular analysis of all alarms should be expected
 - Compare with a surveillance camera.
 - Not much use without a guard watching it
- ❑ Firewall/IPS
 - Traffic does or does not pass the firewall
 - Active filtering
 - Fail-close
- ❑ Network IDS
 - Traffic always passes initially
 - Passive monitoring
 - Fail-open

Known IDS Concerns

- ❑ Inaccuracy of exploit-based signatures
- ❑ Cannot recognize unknown anomalies or intrusions
- ❑ Cannot provide measurable information for forensics or situational-aware analysis, except in case of DPI
 - Difficult to differentiate malicious events with unintentional anomalies
 - Anomalies caused by network element faults, e.g., router misconfiguration or link failures
 - Anomalies caused by application, e.g., P2P misconfigurations
 - Situational awareness not determined
 - E.g., as for an attack scope, an attack target, even the strategy, or an attacker's size for a botnet

Popular Research Area: Alert Handling

- **Alert Aggregation**
 - Simple alerts aggregated within a sensor
- **Alert Correlation**
 - More complex correlations, often between different types of sensors.
 - Interoperability with IDMEF
- **Topology Vetting**
 - By knowing the topology of the network and which services are installed, we can avoid harmless alerts.
- **Alert Prioritizing**
 - Prioritize alerts based on the interest of the operator.



© 2024 UZH, CSG@IFI

49



Takeaways

- **State-of-the-art:**
 - Catches well-known attacks
 - May even catch simple variants.
 - Do not detect
 - new attacks,
 - larger variants, and
 - the Insider (basically ignored).
 - Especially true, if masquerades: uses a little bit more than he should of all resources.
 - False alarms are a major problem.
 - Poorly defined rules
 - Mostly misuse systems, and very few anomaly detection systems among the commercial systems.
- **Research Directions:**
 - AI & IDS Revival
 - More self-aware systems that can self-heal
 - Anomaly-based systems
 - Formally define systems and their detection capabilities
 - Handling gigabit/terabit speeds
 - Alert handling
 - Less of black box solutions

© 2024 UZH, CSG@IFI

50



Computer Network and Security Principles (CNSP)

L07 – Security Information and Event Management (SIEM)

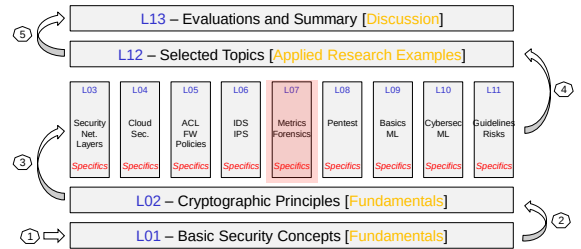
© 2024 UZH, CSG@#

1

ifi

1

Overview of Class and Lectures (Chapter Dependency)



© 2024 UZH, CSG@#

2

ifi

2

Security Information and Event Management

- Objectives
 - To get hold of current landscape of management interfaces in IT security
 - To introduce SIEM systems
 - To overview the concept of security metrics
 - To have a practical overview
- Topics
 - IT security landscape and management complexities
 - Definition of SIEM and deployment types
 - Major capabilities
 - Correlation rules, alerts, dashboards
 - Security metrics and pitfalls
 - Comparison of SIEM tools

© 2024 UZH, CSG@#

3

ifi

3

INTRODUCTION

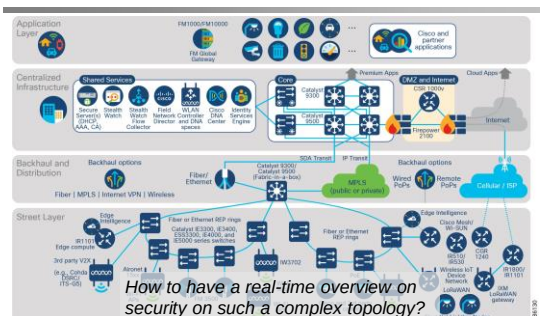
© 2024 UZH, CSG@#

4

ifi

4

ISP Topologies are complex...



© 2024 UZH, CSG@#

5

Cisco Illustration for Connected Communities Infrastructure (C)

ifi

5

IT Security Landscape (1)

- IT Infrastructures are becoming increasingly complex to be managed
 - Many different resources
 - Servers (compute), databases, network equipment
 - And many different services:
 - Access control management, intrusion prevention and detection, event collection, compliance mapping, reporting, ...
- Events are typically reported independently and in real-time
 - For example:
 - A DNS resolution failure may incur in the failure of different services trying to access external sites
 - How to quickly identify the root cause of an(y) issue to restore services?



Swisscom SOC, Switzerland (January, 2010)

© 2024 UZH, CSG@#

6

ifi

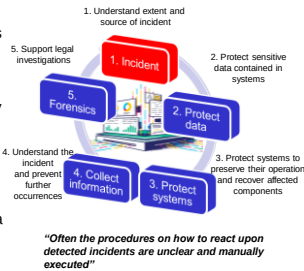
6

IT Security Landscape (2)

- Scenario is even more challenging for **SMEs***:

1. Low cybersecurity awareness
2. Inadequate protection for critical and sensitive data
3. Budgetary issues
4. Shortage of adequate security staff
 - Missing staff and expertise
5. Missing suitable guidelines
6. Inadequate infrastructure (hardware and software) to process large amounts of data
7. Moving online
8. Low management support

SME: Small and Medium-sized Enterprises (between 10-250 employees)



© 2024 UZH, CSG@#

7

*ENISA: [Cybersecurity for SMEs](#), Challenges and Recommendations, 2021



Cybersecurity Awareness

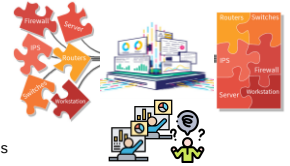
- The key question is how to map threats (and risks) considering data from several of events and data being exchanged in real-time?

– What information is key [to be protected] from a business perspective?

– How to correlate events from different systems?

– What KPI/metrics are relevant to be tracked?

- Take business decisions
- Take operational decisions
- Support of operational decisions



© 2024 UZH, CSG@#

Key Performance Indicator (KPI)

8

Source: <https://nmap.org/using-nmap.html>



SECURITY INFORMATION AND EVENT MANAGEMENT (SIEM)

© 2024 UZH, CSG@#

9



SIEM and Metrics

- Security Information and Event Management (SIEM)
 - Application that provides the ability to **gather security data from information system components** and **present that data as actionable information** via a single interface.



- Security metrics

– Quantifiable measurements used to understand the status of systems and services through the collection, analysis, and reporting of relevant data.

– Based on security objectives that help inform decisions on how to improve the security of all components involved in delivering services and processing data.



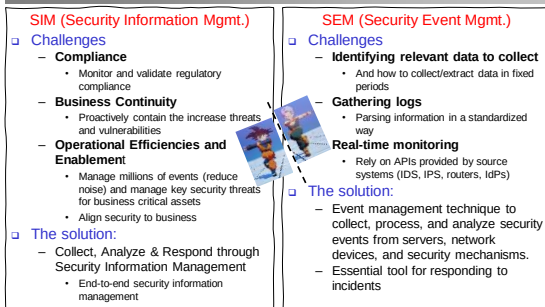
https://csrc.nist.gov/glossary/term/security_information_and_event_management_tool
<https://www.gartner.com/en/information-technology/glossary/security-metrics>

© 2024 UZH, CSG@#

10



SIM and SEM



© 2024 UZH, CSG@#

11



Security Information and Event Management (SIEM)

- SIEM was a term coined in a Gartner report in **2005** as a technology that combines:

– **SIM (Security Information Management)**

- Centralized management, collection, preservation of historic log data and the generation of reports for compliance purposes

– **SEM (Security Event Management)**

- Threat management, real-time monitoring of security incidents and triggering proper reactions in case of an incident

- SIEM evolved from a **sole combination to a more holistic and integrated security solution**

- Support historic analysis of events on a broad range of events, forensics
- Support for Cybersecurity AI/ML (stay tuned for the next lectures)



© 2024 UZH, CSG@#

12



SIEM Objectives

SIEM applies determined procedures to:

- Identify threats
 - Relying on IDS/IPS systems
- Collect logs and context data
- Determine possible data breaches
- Pre-process logs
 - Parsing, normalization, categorization, enrichment
- Perform analysis
 - Correlations, scoring, prioritization
- Run forensic investigations
 - Provide evidences
- Alert stakeholders
 - Security reports



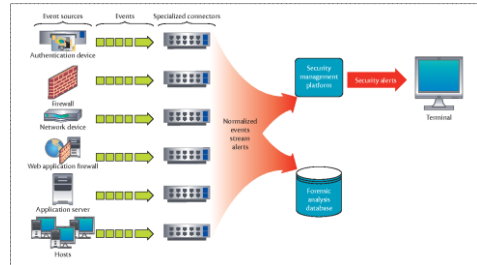
**Naturally SIM and SEM challenges are also combined, making SIEM more complex to be achieved.*

© 2024 UZH, CSG@#

13

ifi

General Information Flow



© 2024 UZH, CSG@#

14

Shen, S., Mandelblat, P., and Dinstein, I., 2014. The Operational Role of Security Information and Event Management Systems. IEEE security & Privacy, 1(2), pp.35-41.

ifi

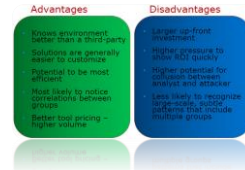
Deployment Types

Outsourced, i.e., cloud-based



vs.

In-house



© 2024 UZH, CSG@#

15

ifi

Major Capabilities

- Data acquisition and aggregation:** SIEM solutions aggregate data from many sources, including network, security, servers, databases, applications, providing the ability to consolidate monitored data.
- Event correlation:** looks for common attributes, and links events together into meaningful bundles. This technology provides the ability to perform a variety of correlation techniques to integrate different sources, in order to turn data into useful information.
- Alerting:** the automated analysis of correlated events and production of alerts, to notify recipients of immediate issues.
- Dashboards:** SIEM tools take event data and turn it into informational charts to assist in seeing patterns or identifying activity that is not forming a standard pattern.
- Compliance:** SIEM applications can be employed to automate the gathering of compliance data, producing reports that adapt to existing security, governance and auditing processes.
- Retention:** SIEM solutions employ long-term storage of historical data to facilitate correlation of data over time, and to provide the retention necessary for compliance requirements.



© 2024 UZH, CSG@#

16

ifi

Data Acquisition and Aggregation

Sources (logs and protocols)

- Event Collector and Processor
 - Syslog, Log Files, SMB, ODBC, All Log Sources
- Flow Collection and Processor
 - NetFlow, J-Flow, S-Flow, IPIX
- Asset Database
 - Based on Asset Criticality, Risk and Vulnerability, System and Business Owner
- Event and Flow Correlation
 - Advanced Threat Analytics
- Centralized Management Console for Security Dashboard and Reporting
- Integration with service desk for automated ticket creation
 - Offense Management

Network and Security Monitoring (real-time data)

- Network Performance Monitor
 - SNMP, OpenFlow
- Network Monitoring
 - Link Utilization Rate
- Availability Monitoring
 - Average resource usage, CPU/RAM, Disk
- SLA reporting
- Integration with service desk for automated ticket creation



© 2024 UZH, CSG@#

17

ifi

Real-time Event Correlation

Real-time may mean

- seconds or hours – depending on the application

Correlation means

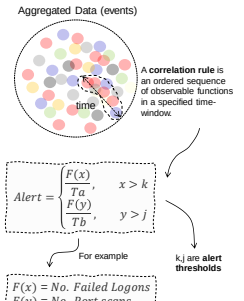
- the dependence on **any statistical relationship**, causal or not, between two or more random variables or bivariate data

Impacts

- (Pro-)actively addressing security threats
- Simultaneous processing of events to detect anomalies

Basis due to

- Log search
- Customized rules and alerts



© 2024 UZH, CSG@#

18

ifi

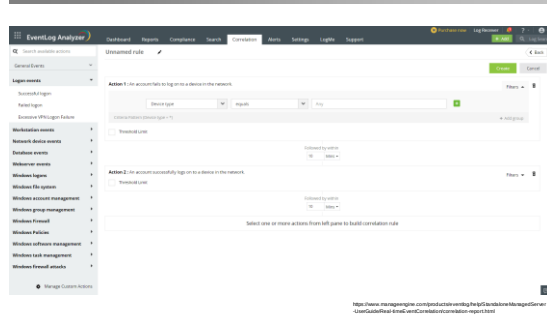
Correlation Rules

- | | |
|--|---|
| <p>➤ Correlation Rule: Possible ransomware activities (Windows)</p> <ul style="list-style-type: none"> – A ransomware can be identified with a process being started, shortly followed by multiple file modifications. <p>➤ General action pattern: Failed logon -> Failed logon -> File modified -> (...) -> File modified (all within a few minutes, on the same device)</p> <p>➤ Specific action pattern: Process started -> (within the next 5 minutes) -> At least 15 file modifications on the same device, by the same process</p> <p>➤ The rule can thus be configured as below:</p> <p>Action 1: Windows process started - a process is started on Windows.</p> <p>Time window between Action 1 and Action 2: 5 minutes</p> <p>Action 2: File modified - a file is modified on a Windows device.</p> <p>Threshold: 15 times within 30 minutes.</p> <p>• Filters:</p> <ul style="list-style-type: none"> • The device name should be the same for all occurrences of Action 2. • The process name should be the same for all occurrences of Action 2. • The device name should be the same as the device name from Action 1. • The process name should be the same as the process name from Action 1. | <p>➤ Correlation Rule: Brute force</p> <ul style="list-style-type: none"> – A brute force attack is characterized by several failed logons on a device, followed by a successful logon. <p>➤ General pattern: Failed logon -> Failed logon -> Failed logon - (...) -> Successful logon (all within a few minutes, to the same device)</p> <p>➤ Specific pattern: At least 10 failed logons to a single device within 2 minutes -> (within the next 1 minute) -> Successful logon to the same device</p> <p>➤ The rule can thus be configured as below:</p> <ul style="list-style-type: none"> – Action 1: Failed logon - an employee fails to log on to a network device. – Threshold: This action should occur an minimum of 10 times within 2 minutes. <p>Filters: The device name should be the same for all occurrences of Action 1.</p> <p>Time window between Action 1 and Action 2: 1 minute</p> <p>Action 2: Successful logon - an employee logs on to a network device.</p> <p>Threshold: None.</p> <p>Filters: The device name should be the same as the device name from Action 1.</p> |
|--|---|

**ifi**

19

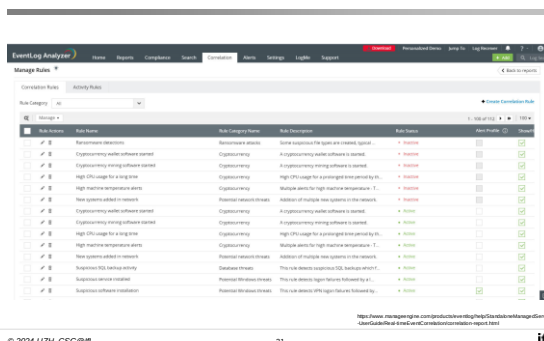
Creating a Correlation Rule



if

20

Visualizing Correlation Rules

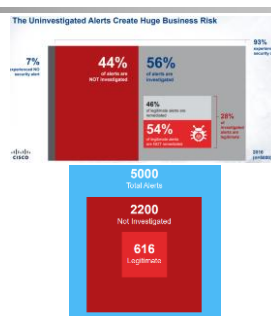


ifi

21

Security Alerts (1)

- ❑ A Cisco [survey](#) (2017) of large companies determined that 44% of organizations were seeing an average of 5,000 alerts daily.
 - Less than 60% of the alerts were investigated.
 - Even though 28% of the alerts were cited as legitimate concerns to information security.
- ❑ **Tuning alerts** is a way reduce irrelevant and duplicate alerts, and false positives.
 - Alerts without context and priority levels are also difficult to focus on.



if

22

Security Alerts (2)



- ❑ An alert define the **conditions (thresholds)** to be observed in a **security metric**
 - Indicators of Attack (IoA)
 - Indicators of Compromise (IoC)
- ❑ **Typically expressed in a dashboard**
 - Depends on data (logs/events) available
 - Key to express correlation rules (combining two or more alerts)
- ❑ **Selected points to ensure efficient alerts:**
 - Review rules and alert configurations periodically
 - Consider an updated analysis of risks to refine alerting rules
 - Keep an inventory of systems, apps, and privilege changes
 - Reduce alerts whenever possible
 - Define categories to prioritize alerts (e.g., critical, high, medium, low)
 - Determine behavioral baselines
 - Systems (and security) are constantly updated/patched

$$\text{Alert} = \left\{ \begin{array}{l} \frac{F(x)}{Ta}, \\ \frac{F(y)}{Tb} \end{array} \right\} \quad \begin{array}{l} x > k \\ y > j \end{array}$$

Metrics Thresholds

ifi

23

Dashboards (1)



- ❑ **User interface**
 - Presentation of results to experts and non-experts
 - Visual display of information
 - Reports and statistics
 - Overview of alerts
- ❑ **Separate management interfaces**
 - Alerts, correlation rules, responses
- ❑ **Context or meta data does add on**
 - Identifying more precise event or action details
 - Determining the involvement of stakeholders
 - Correlations with past knowledge and historical background



Splunk dashboard

if

24

Dashboards (2)



© 2024 UZH, CSG@#

25

IBM QUTUM LAB PROOF OF CONCEPT ifi

Dashboards (3)



© 2024 UZH, CSG@#

26

AlertVault OSSIM ifi

Compliance



- Organizations must **comply with regulations** defined by regulatory bodies
 - General Data Protection Regulation (GDPR)
 - Health Insurance Portability and Accountability Act (HIPAA)
 - Payment Card Industry (PCI) compliance
 - Insider threats (access to customer data, breaches, abuses)
 - Threat identification and hunting
- SIEM is critical to **implement controls** to ensure:
 - Data privacy policies**
 - Proof state of security on stored data
 - Control frameworks**
 - Discovery and continuous monitoring of usage
 - Verify compliance policies, attempted data alterations
 - Internal audits**
 - Failed application usages
 - Login attempts
 - Transactions verification



<https://www.dnsstuff.com/common-siem-alerts>

© 2024 UZH, CSG@#

27

ifi

Retention



- Data retention**
 - Depending on **legal** situation: short- vs. long-term of raw data
 - Also depends on compliance
 - Log collection for all logs of raw data
 - Highly scalable
 - Storage optimized
- Challenges**
 - Compliance of log data handling with legal constraints
 - Supporting the basis for an open root cause analysis
 - Analysis of event logs for security incidents
 - Handling vast amounts of possibly incompatible data
 - Tracking suspicious user behavior
 - Unsecure data storage



<https://www.blumira.com/10-siem-log-sources-to-prioritize/>

© 2024 UZH, CSG@#

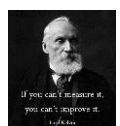
28

ifi

SECURITY METRICS

Metrics and Attributes

- Metric is a **group of measurements that produce a quantitative picture of something over a period**
 - Metrics are specific, measurable, attainable, repeatable, and time-dependent
- Objectives**
 - Collect** objective information about the network's security
 - Track** progress toward improvement goals
 - Assess** the impact of process/tools changes
- Good metrics vs metrics**
 - Quantitative
 - Objective
 - Based on a formal model
 - Has a time dimension
 - Universally acceptable
 - Has ground truth
 - Inexpensive
 - Obtainable
 - Repeatable



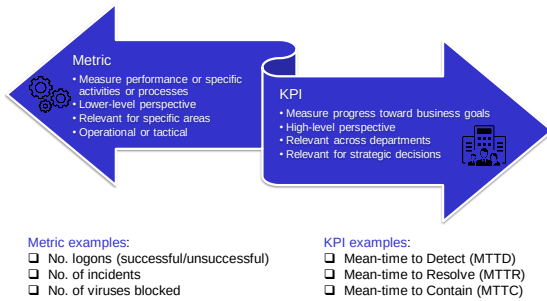
Yee, G. O. (2013). Security Metrics: An Introduction and Literature Review. Computer and Information Security Handbook, 553-566.

© 2024 UZH, CSG@#

30

William Thomson, Lord Kelvin ifi

Metric vs KPI



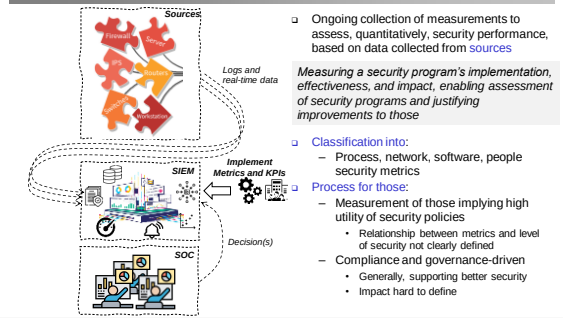
© 2024 UZH, CSG@ifi

31

ifi

31

Security Metrics



© 2024 UZH, CSG@ifi

32

ifi

32

Examples at Different Levels

Process

- No. of systems with risk assessments, no. if risks identified, no. of policy violations, percentage of weak passwords in use, percentage of systems with contingency plans

Network

- Traffic analysis, (un-)successful log-ins, no. of incidents, no. of viruses blocked, no. of patches applied, no. of spam blocked, no. of virus infections, no. of port probes

Software

- Size and complexity, defects/LoC, defects over time, attack surface, security layers, design flaws

People value between reliability and security

© 2024 UZH, CSG@ifi

33

ifi

33

“Measuring” Security in Reality

Initially: Trusted Computer System Evaluation Criteria (TCSEC) DoD standard (1983)

- ITSEC (Europe)
- CTCPEC (Canada)

Today: Common Criteria for Information Technology Security Evaluation (since 1996)

- Secured in ISO/IEC 15408 (1999)

NIST FIPS-140 series (2001)

- Design of cryptographic modules and their requirements

NIST SP 800-55 (2008)

- Performance Measurement Guide for Information Security

© 2024 UZH, CSG@ifi

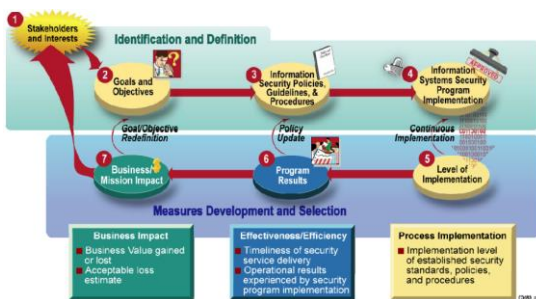
34

<https://csrc.nist.gov/publications/detail/sp/800-55/rev-1/final>

ifi

34

Measures Development Process



© 2024 UZH, CSG@ifi

35

NIST: Performance Measurement Guide for Information Security, NIST SP 800-55

ifi

35

Major Pitfalls Defining Metrics

- Lack of management commitment
 - Measuring too much, too soon
 - Measuring too little, too late
 - Measuring the wrong things
 - Imprecise metrics definitions
 - Using metrics data to evaluate individuals
 - Using metrics to motivate, rather than to understand
 - Collect data that is not used
 - Lack of communication and training
 - Misinterpreting metrics data
- Operational issues**
- (mis-)Management issues**

Lowans, P. W. (2002). Implementing a Network Security Metrics Program. GIAC Administrative Version, (2.0).

© 2024 UZH, CSG@ifi

36

ifi

36

COMPARISON OF SIEM TOOLS

© 2024 UZH, CSG@#

37

ifi

37

Selected List of Tools and Vendors

- ArcSight Enterprise Security Manager
 - MicroFocus/HPE/NetIQ
- Qradar
 - IBM
- McAfee Enterprise Security Manager
 - McAfee/Intel
- LogRhythm Next GEN SIEM Platform
 - LogRhythm
- USM and OSSIM
 - AT&T Cybersecurity/ AlienVault
- RSA Netwitness Platform
 - Dell
- Splunk Enterprise Security
 - Splunk
- SolarWinds Log and event Manager
 - SolarWinds



© 2024 UZH, CSG@#

38

ifi

38

Features and Capabilities

Capabilities	ArcSight	QRadar	McAfee	LogRhythm	USM-OSSIM	RSA	Splunk	SolarWinds
Correlation rules	●	●	●	●	●	●	●	●
Data sources	●	●	●	●	●	●	●	●
Real-time processing	●	●	●	●	●	●	●	●
Data volume	●	●	●	●	●	●	●	●
Visualization	●	●	●	●	●	●	●	●
Data analytics	●	●	●	●	●	●	●	●
Performance	●	●	●	●	●	●	●	●
Forensics	●	●	●	●	●	●	●	●
Complexity	●	●	●	●	●	●	●	●
Scalability	●	●	●	●	●	●	●	●
Risk analysis	●	●	●	●	●	●	●	●
Storage	●	●	●	●	●	●	●	●
Price	●	●	●	●	●	●	●	●
Resilience	●	●	●	●	●	●	●	●
Reaction and reporting	●	●	●	●	●	●	●	●
UEBA	●	●	●	●	●	●	●	●
Security	●	●	●	●	●	●	●	●



User and Entity Behavior Analytics (UEBA)

Correlation Rules, G. Correlation Rules, S. & D. R. (2023). Security Information and Event Management (SIEM), Analysis, Trends, and Usage in Critical Infrastructures, Sensors, 21(2), 479.

© 2024 UZH, CSG@#

39

ifi

39

Gartner Analysis (as of 2022)



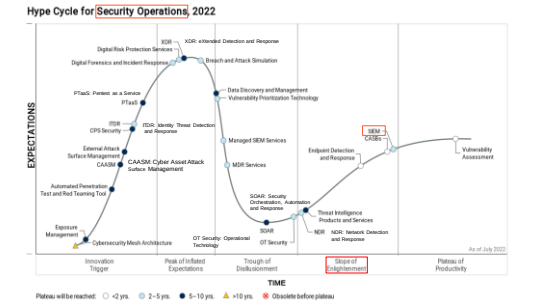
© 2024 UZH, CSG@#

40

ifi

40

Gartner Hype Cycle (as of 2022)



© 2024 UZH, CSG@#

41

ifi

41

Gartner Hype Cycle (as of 2023)



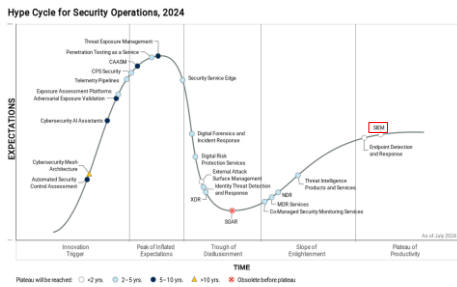
© 2024 UZH, CSG@#

42

ifi

42

Gartner Hype Cycle (as of 2024)



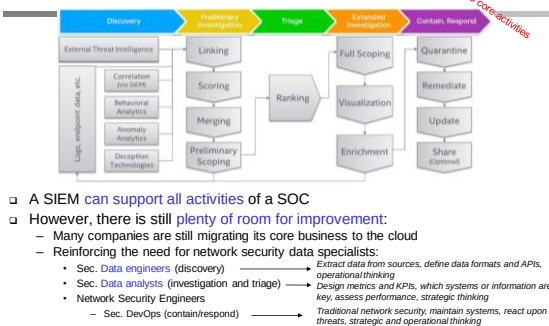
TAKEAWAYS

Need for Standardized Solutions

- ❑ Rise of number of data breaches dramatic
 - Pressing factor
 - Internal and external threats
- ❑ Standard security tools may be too traditional
 - Smart attackers and new metrics
- ❑ Broad number of log sources to be managed
 - Multiple and distinct devices generate log data
- ❑ Legal and partially company-internal guidelines
 - Compliance regulation(s) to be met
- ❑ Cyberattack mitigation
 - If possible, correct and fast decisions needed to mitigate

- ❑ Limited quality of information
 - Although capturing large amounts of data, still not possible to capture all incidents.
 - Identity data is often fragmented (GDPR reasons)
 - Not cost-effective to capture and process all the required data
 - Limited storage capabilities
 - A major challenge in researching and developing new alert rules is creating effective strategies to identify system attacks and anomalies
- ❑ Basic correlation rules
 - Imprecise or pointless security metrics
 - Performance tradeoff involving basic vs complex correlation rules
- ❑ Reliance on human input
 - And we make mistakes...
- ❑ SIEMs typically do not provide high-level security risk metrics (i.e., KPIs)
 - Still lack the development of useful operational metrics to support decisions based on quantitative evidence
 - Not only quantitative, but also probabilistic metrics
- ❑ Basic reaction and reporting/compliance
 - Revolves to the limited quality of information, imprecise metrics, basic correlation rules, and human mistakes
 - Not always clear how to respond to an incident
 - Complex to automate commands across different devices → lack of standard interfaces
- ❑ Still limited research on the topic
 - Most references are industry-based

SIEM and SOC



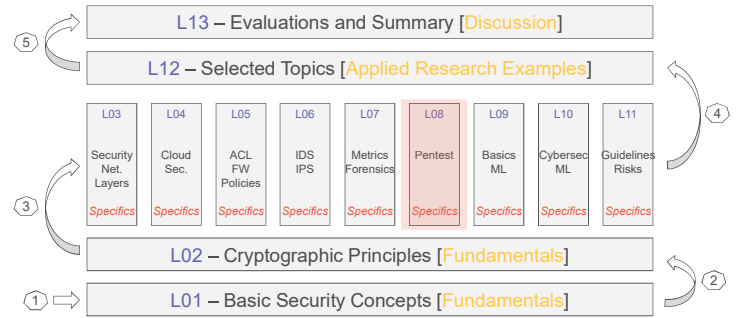
Takeaways

- ❑ Metrics are the key enabler to improving your network's security.
 - You cannot improve what you cannot measure
 - Key is defining which set of metrics and KPIs are essential to ensure internal/external compliance
- ❑ SIEM solutions are still on the rise of their adoption
 - Define a centralized interface to manage security of complex environments.
 - Lack of standards: Different tools, different data formats, different protocols, interfaces
 - Difficult for large companies, even more difficult for SMEs
 - Limited scalability, high complexity to calculate metrics, alerts, correlation rules, poor usability
- ❑ Data-oriented security is a reality, but far to reach its potential
 - Technology-oriented solutions tend to fail as requirements evolve, but data is always at core
 - Machine-learning approaches to improve detection and automate reaction

Computer Network and Security Principles (CNSP)

L08 – Penetration Tests

Overview of Class and Lectures (Chapter Dependency)



Penetration Test

Objectives

- To get hold of the concept of penetration testing and its tasks
- To introduce a general “pentest” methodology
- To overview major tools used in different methodology steps
- To have a practical overview by an industry expert

Topics

- Pentest Motivation and definition
- Types and existing methodologies
- General methodology
- Summary and takeaway's
- Pentest practical introduction

BASICS

Why Pen Testing?

- Secure critical «web-facing» applications such as in the financial sector (banks, investment firms) or exposing personal information (storage clouds)

- Identify potential vulnerabilities that a malicious user could exploit.
- Validate the security of an organization's systems, applications, and networks

- It helps to build:

- Secure infrastructure
- Customer trust and company reputation
- Efficient security measures and security awareness



<https://www.ibm.com/security/data-breach>

Data breach average cost increased 2.6% from USD 4.24 million in 2021 to USD 4.35 million in 2022. The average cost has climbed 12.7% from USD 3.86 million in the 2020 report.

Stolen or compromised credentials were responsible for 19% of breaches. Phishing was responsible for breaches 16% of the time. Cloud misconfiguration caused 15% of breaches.

Definition

- Penetration testing, also known as “ethical hacking”, is formally defined as:

“...a legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making those systems more secure. The process includes probing for vulnerabilities as well as providing proof of concept attacks to demonstrate the vulnerabilities are real.” (pp. 1, Engebretson, 2013)

Goal:

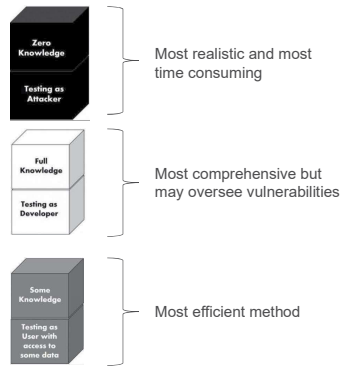
- Improve the security of the target system
- Thus, “ethical hacking”

Two key elements in the definitions:

1. It is an authorized attempt
2. It uses a non-exhaustive set of tools and techniques to probe for vulnerabilities

Types of Pen Test

- ❑ **Black-box**
 - Tester has **no knowledge** of the system rather than who the target is.
 - Useful to evaluate external security.
- ❑ **White-box**
 - Tester has **full knowledge** and may work closely with the target
 - Useful to evaluate internal security.
- ❑ **Gray-box**
 - A **mixture of closed/open-box** types with no general rule to define a «gray» area.
 - Useful to evaluate/understand the security specific components, limiting the impact on components clearly out of the scope



Existing Methodologies

- ❑ Penetration tests can deliver **widely different results** depending on which standards and methodologies they leverage
- ❑ **Similar but not the same:**
 - **Standard or guideline**
 - PTES
 - PCI
 - NIST 800-115
 - **Framework**
 - MITRE ATT&CK
 - ISSAF
 - Penetration Testing Framework
 - **Manual**
 - OSSTMM
- ❑ OWASP Testing Guide
- ❑ PCI Penetration Testing Guide
- ❑ MITRE ATT&CK
 - Widely used in industry
- ❑ Penetration Testing Execution Standard (PTES)
- ❑ NIST 800-115
- ❑ Penetration Testing Framework
- ❑ Information Systems Security Assessment Framework (ISSAF)
- ❑ Open Source Security Testing Methodology Manual (OSSTMM)

Example/Differences

- ❑ **PTES** defines penetration testing as 7 key phases.
 1. Pre-engagement Interactions
 2. Intelligence Gathering
 3. Threat Modeling
 4. Vulnerability Analysis
 5. Exploitation
 6. Post Exploitation
 7. Reporting
- ❑ **MITRE ATT&CK** includes twice more key sections:
 1. Reconnaissance
 2. Resource Development
 3. Initial Access
 4. Execution
 5. Persistence
 6. Privilege Escalation
 7. Defense Evasion
 8. Credential Access
 9. Discovery
 10. Lateral Movement
 11. Collection
 12. Command and Control
 13. Exfiltration
 14. Impact

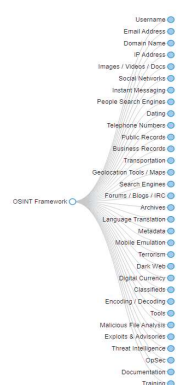
GENERAL METHODOLOGY

Pentest Methodology

- ❑ There are a number of available methodologies **typically** involving the steps (Engebretson, 2013):
 1. **Intelligence Gathering** (recoinnaisance)
 - Understanding and collecting passive information about the target system and processes.
 2. **Mapping/Scanning**
 - Active scanning of functionalities and weaknesses to assess the target.
 3. **Vulnerability Identification** (discovery)
 - Discovery of vulnerabilities/exploits complementing automated tools with manual searches
 4. **Exploitation**
 - Testing mapped vulnerabilities/flaws to discover new vulnerabilities
 5. **Reporting**
 - Document all findings

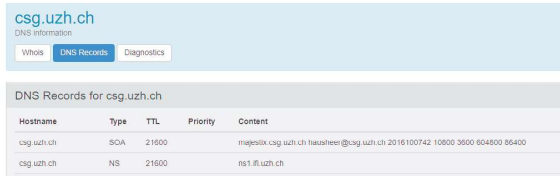
1. Intelligence Gathering/ Reconnaissance

- ❑ **Major preparation/planning step**
- ❑ **Goal:**
 - Reconnaissance against a target to gather as much information as possible
- ❑ **Targets/entry points** can be:
 - Physical (critical infrastructures), electronic (websites), human (phishing, crypto wallets). Also categorized as *organizational* or *individual*
- ❑ **Can be classified as:**
 - **Active:** engage directly with the target
 - **Passive:** use of search engines, social media, and other websites to gather information about the target
- ❑ **Open Source Intelligence (OSINT) Gathering**
 - Approach and collection of tools to gather open information



Whois/DNS

- Whois holds a huge database that contains information regarding almost every website that is on the web
- <https://who.is/>
 - Check for DNS records, which services are hosted (possible open ports)



DNS Records for csg.uzh.ch				
Hostname	Type	TTL	Priority	Content
csg.uzh.ch	SOA	21600		masterix.csg.uzh.ch. hakuheri@csg.uzh.ch 2016100742 10800 3600 604800 86400
csg.uzh.ch	NS	21600		ns1.f.uzh.ch

2. Mapping/Scanning

- Goal:** Understand how the target application will respond to various intrusion attempts
 - Active probing the target
 - Can also be seen as part of "Intelligence Gathering" and vice-versa
- Methods:**
 - Host discovery (network/port scanning)
 - Scanning programs that identify live hosts, open ports, services, and other info (Nmap, autoscanner)
 - DNS Querying (nslookup, dig)
 - Route analysis (traceroute)
 - OS Detection
 - Bypassing firewalls

The targeted organization may become aware of the ongoing process since there is an active engaging with the target. At this stage, we obtain information about the open ports, services, versions of the applications, version of the operating system

Host Discovery

- The first step of a network pentest most times would be to know what targets are alive.

```
Pinging www.google.com [74.125.232.145] with 32 bytes of data:
Reply from 74.125.232.145: bytes=32 time=25ms TTL=51
Reply from 74.125.232.145: bytes=32 time=198ms TTL=51
Reply from 74.125.232.145: bytes=32 time=245ms TTL=51
Reply from 74.125.232.145: bytes=32 time=165ms TTL=51

Ping statistics for 74.125.232.145:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 165ms, Maximum = 253ms, Average = 215ms
```

- Also possible to scan network ranges with nmap

```
root@root:~# nmap -sP 192.168.15.1/24

Starting Nmap 5.51 ( http://nmap.org ) at 2013-06-09 16:16 EDT
Nmap scan report for WlMaxCPE (192.168.15.1)
Host is up (0.0026s latency).
MAC Address: 20:10:7A:BF:AA:4B (Unknown)
Nmap scan report for root (192.168.15.14)
Host is up.
Nmap scan report for Princydule-PC (192.168.15.159)
Host is up (0.0036s latency).
MAC Address: 00:24:D6:60:1A:9C (Intel Corporate)
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.53 seconds
```

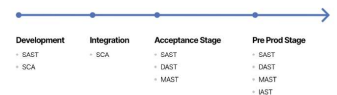
External scanning is often blocked by IDS/IPS systems

3. Vulnerability Scanning

- Goal:** identify and evaluate the security risks posed by identified vulnerabilities. Vulnerability analysis work is divided into two areas: identification and validation.
- Vulnerabilities are often associated with missing patches.
 - Unpatched systems often lead to remote code execution
 - Vendors often release patches to fix a known problem or vulnerability.
- Application Security Testing (AST)**
 - Static (SAST),
 - Dynamic (DAST)
 - Interactive (IAST)
 - Run-time (RASP)

Remote code execution allows an attacker or penetration tester to fully and completely control the remote computer as if he/she were physically sitting in front of it. This includes, but is not limited to, copying, editing, and deleting documents or files, installing new programs, making changes or disabling defensive products like firewalls and antivirus, setting up key loggers or backdoors, and using the newly compromised computer to attack new machines.

- Major AST types are associated with different pentest types at different stages of development:
 - SAST → whitebox
 - DAST → blackbox
 - IAST → graybox



Vulnerability Scanning Tools

- Web Application Vulnerability Scanners are automated tools that scan web applications (DAST)
 - List of tools by OWASP: https://owasp.org/www-community/Vulnerability_Scanning_Tools
- OpenVAS - Open Vulnerability Assessment Scanner
- Nessus is a widely used tool (commercial):
 - Scan and report
 - Free up to 16 Ips



Item	CVE	Plugin	Severity	Score
1	2013-06-09	2013-06-09	High	9.0
2	2013-06-09	2013-06-09	High	8.8
3	2013-06-09	2013-06-09	High	8.8

4. Exploitation

- Goal:** The exploitation phase of a penetration test focuses solely on establishing access to a system or resource by bypassing security restrictions.
- Wide range of possibilities, approaches, tools to exploit a vulnerability
 - Each system is different and each target is unique (different OS, services)
 - Each tester/attacker is also unique in their set of skills/preferences
- Divided between remote and client-side exploitation
 - Remote: direct contact with server without user interaction
 - Client: engage with the target to exploit

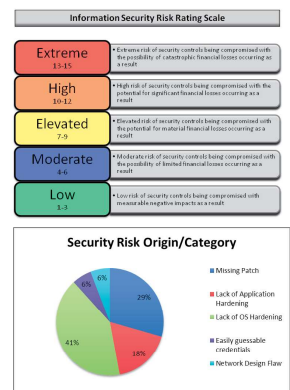
Metasploit

- ❑ One of the most powerful tools for penetration testing.
 - Free and open source
 - <https://www.metasploit.com/>
- ❑ Systematically probe a target for vulnerabilities
 - Enables to write, test, and execute exploits
- ❑ Several tutorials available



5. Reporting

- **Goal:** define the base criteria for penetration testing reporting
- At minimum, a report should include the following:
 - An executive summary.
 - A walkthrough of how the penetration test was performed to provide an understanding of how you successfully compromised or hacked the system(s).
 - A detailed report.
 - Raw output (when requested) and supporting information.



Report's Structure

Executive Summary

- ❑ Very **brief overview of major findings** not exceeding two pages
- ❑ **Does not provide technical details** or terminology
 - Should target board members and nontechnical management people
- ❑ If vulnerability and exploits were discovered, the executive summary needs to focus on **explaining how** these **findings impact the business**

Detailed Report

- ❑ Comprehensive list of findings as well as the technical details
 - Target IT managers, security experts, network administrators, and others who possess the skills and knowledge required

Raw Output

- ❑ Complements the detailed report by providing the raw output from each tool
 - Custom tools, scripts, or proprietary code are typically not included
- ❑ Allow to reproduce the steps to exploit vulnerabilities

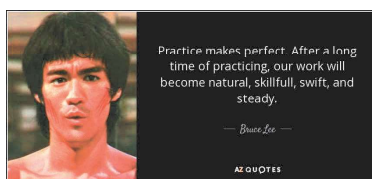


Summary

- Pentest is one of the most important activities to secure an organization's infrastructure
 - Allows to assess risks, check whether regulations and compliance are met, build/maintain reputation with customers
- There is a wide range of approaches, ranging from methodologies, frameworks, guidelines, manuals
 - Important to check which ones fits better to an organization
- Although having well defined methodologies, penetration testing is an activity that requires practice and experience to be improved over time
 - Highly diverse types of targets, systems and services
 - The sole use of tools is not impactful without knowledge

Takeaways

- ❑ Learn networking basics
 - Highly important to learn basic protocols and systems
- ❑ Learn a structured sequence of steps, i.e., methodology/framework
 - ...but have in mind that experience helps to adjust the use of strategies and tools on the fly



INTRODUCTION



- **Cyber Security Journey Start**
- **What is Penetration Testing?**
 - Pen-Tester
 - Vulnerability Analysis
 - Red Team
- **Purpose of Today's Talk**
 - Working as a Pen-Teseter
 - Tools, Skills, & Challenges
 - Live Demo

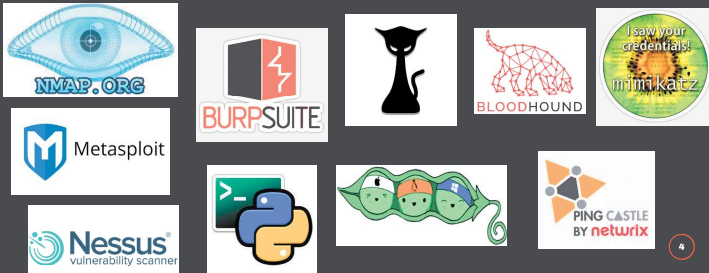
2

DAY-TO-DAY LIFE OF A PENTESTER

- Planning & Scoping**
Three important meetings with the client
- Collaboration**
Client IT, Internal IT (SysOps, CybSec)
- Vulnerability Analysis & Exploitation**
Manual and automated tests
- Post - Exploitation**
Domain Admin! Now what?
- Reporting**
Final product

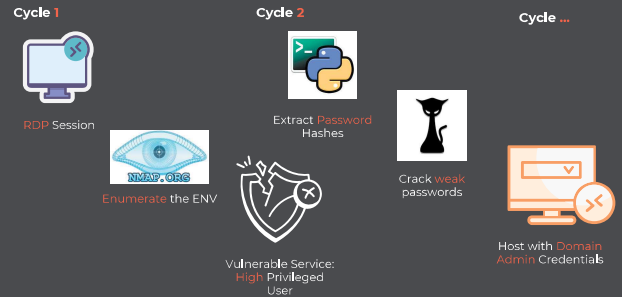
3

TOOLS & TECHNIQUES



4

TYPICAL PENTEST SCENARIO



5

FEELINGS BEHIND THE JOB

- Excitement
- Frustration and Patience
- Satisfaction & Responsibility
- Up to Date



6

SKILLS & TRAITS

- Curiosity & Analytical Mind**
 - Leave no stone unturned; question everything
 - Pay attention to detail
- Technical Knowledge**
 - OS; Networking; Client-Server;
 - Finding an exploit is only half the job
 - Certifications (e.g., OSCP)
- Communication Skills**
 - Manage expectations
 - Various parties involved
 - You might break stuff
 - Report is your final product



7

CHALLENGES

Time Management by Tight deadlines

Short Pen-Tests

Unresponsive Clients

User Accounts & other queries

Scope (limited vs. large ENV)

Few applications

Tools

What to trust?

8

CHALLENGES

Time Management by Tight deadlines

Short Pen-Tests

Unresponsive Clients

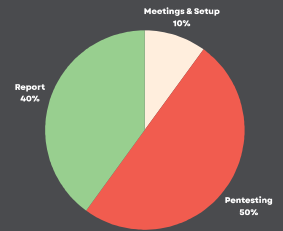
User Accounts & other queries

Scope (limited vs. large ENV)

Few applications

Tools

What to trust?



9

CHALLENGES

Time Management by Tight deadlines

Short Pen-Tests

Unresponsive Clients

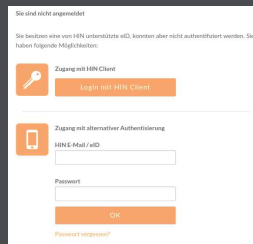
User Accounts & other queries

Scope (limited vs. large ENV)

Few applications

Tools

What to trust?



10

CHALLENGES

Time Management by Tight deadlines

Short Pen-Tests

Unresponsive Clients

User Accounts & other queries

Scope (limited vs. large ENV)

Few applications

Tools

What to trust?



11

CHALLENGES

Time Management by Tight deadlines

Short Pen-Tests

Unresponsive Clients

User Accounts & other queries

Scope (limited vs. large ENV)

Few applications

Tools

What to trust?

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL v2.0. The newest version is always available at; <https://github.com/vanhauser-thc/thc-hydra>
Please don't use in military or secret service organizations, or for illegal purposes. (This is a wish and non-binding - most such people do not care about laws and ethics anyway - and tell themselves they are one of the good ones.)
Example: `hydra -l user -P passlist.txt ftp://192.168.0.1`

12

JOB PATHS

Penetration Tester

Authorized to break in and find vulnerabilities.



SOC Analyst

Monitors and defends the network from live threats.

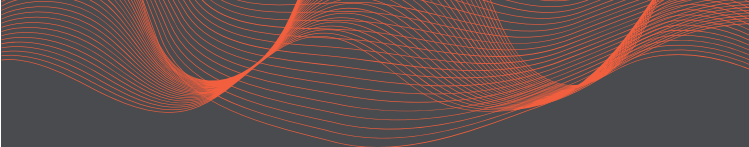


Red Teamer

Simulates a real-world, sophisticated attack to test defenses.



13



Thank you 4 Listening

Computer Network and Security Principles (CNSP)

L9 – Risk Management

© 2024 UZH, CSG@ifi

ifi

1

Risk Management

- ❑ Objectives
 - To understand the concepts and guidelines for risk management from a technical and economic perspective
 - To get insights into well-accepted cybersecurity frameworks and their applications
 - To remember the key concepts of risks and their different impacts
 - Be able to understand and conduct risk management and cybersecurity planning
- ❑ Topics
 - Cybersecurity Domains
 - Overview of Risk Management and Cybersecurity Frameworks
 - Risk Definition and Processes for Risk Management
 - Cybersecurity and Economics
 - Models for Cybersecurity Economics

© 2024 UZH, CSG@ifi

2

ifi

2

Introduction

- ❑ As businesses and governments go digital, they are exposed to **increasing number of threats**
 - Governance, planning, risk assessment, security assessment, and operations management are critical for the digital era
- ❑ Cybersecurity is no longer “just” a technology perspective
 - **Societal and economic** impacts equally important



Cyber attack affects two south England ambulance services
 19 July 2023 (BBC, 2023)
 16 August 2023

© 2024 UZH, CSG@ifi

ifi

3

© 2024 UZH, CSG@ifi

4

ifi

4

Part 1: Risk Management

Central Question

How much money does a company need to invest in order to minimize impact and likelihood of a cyber attack?



What is the optimal investment?

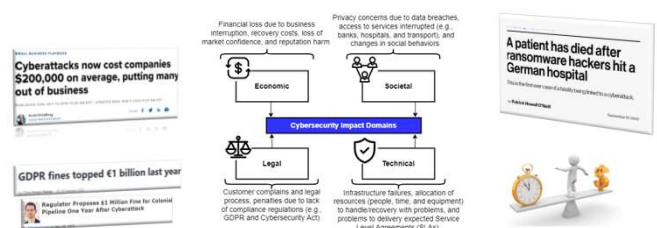
© 2024 UZH, CSG@ifi

4

ifi

5

Cybersecurity Impact Domains



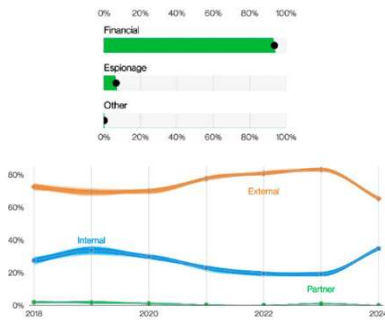
© 2024 UZH, CSG@ifi

6

ifi

6

Motivation Behind Cyberattacks

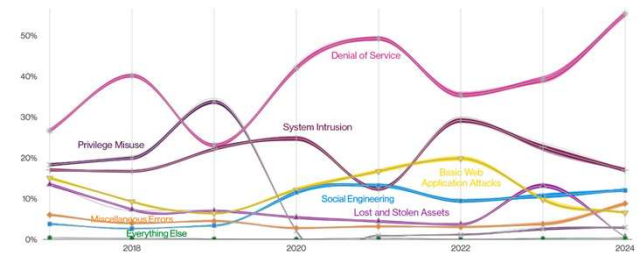


© 2024 UZH, CSG@ifi

7

ifi

Threat Landscape

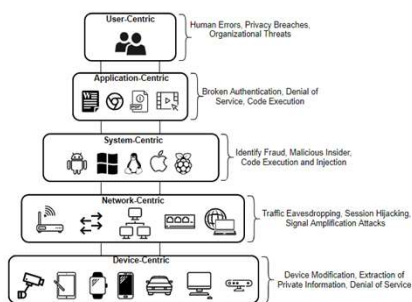


© 2024 UZH, CSG@ifi

8

ifi

Threat Landscape cont.



© 2024 UZH, CSG@ifi

9

ifi

Most Common Threats

Summary of the Most Common Threats According to an ENISA Survey from 2021 within 249 SMEs from 25 European Member States

Threat	Description	Domain	Reported SMEs
Phishing	A digital form of social engineering to deceive individuals into providing sensitive information	User-Centric	41%
Web-based Attack	Exploit of vulnerabilities in code or features of websites to gain access to a server or sensitive information	Application-Centric	40%
Malware	Malicious code that explores vulnerabilities of systems to compromise their functions or act in the benefit of the attacker	Application and System-Centric	39%
Malicious Insider	Legit employees and users that use their access to act maliciously against the company (e.g., steal information or sabotage)	User-Centric	19%
Denial of Service	Interrupt the service or operation of a business by overloading its computer resources with malicious usage	Application, System, and Network-Centric	13%
Social Engineering	Set of techniques to manipulate people to obtain benefits, steal sensitive information, and access critical resources of a company (e.g., buildings, devices, and servers)	User-Centric	11%
Compromised/Stolen Device	Loss of encrypted devices or usage of infected devices by legit employees in the company's systems	Device-Centric	7%

SME: Small- and Medium-sized Enterprise

© 2024 UZH, CSG@ifi

10

ifi

Cybersecurity Guidelines

- Certification requires implementing **best practices** according to cybersecurity standards
- **Planning and management** of cybersecurity are key
 - General frameworks and guidelines
 - Specific documentation for risk assessment and implementation of security controls
- Different institutes and organizations focus on providing **de facto standards for cybersecurity**
 - National Institute of Standards and Technology (NIST)
 - European Union Agency for Cybersecurity (ENISA)
 - International Organization for Standardization (ISO)



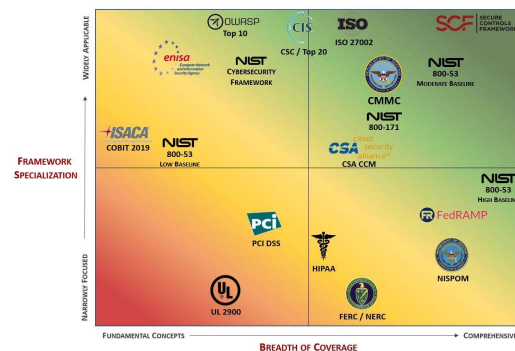
SCF: Security Control Framework

© 2024 UZH, CSG@ifi

11

ifi

Overview of Cybersecurity Frameworks



© 2024 UZH, CSG@ifi

12

ifi

NIST Cybersecurity Framework (CSF)

- High-level scope, focusing on how to implement and prioritize security functions and protections
 - Points to specific documentation, e.g., NIST 800-53 and ISO 27000



© 2024 UZH, CSG@ifi

13

ifi

NIST 800-53

- Regulatory document defining the processes and controls needed to comply with certain security certifications
 - Part of the NIST Risk Management Framework (RMF)
- Divided in three basic control baselines: Low, Medium, and High
 - Set of minimum security controls for different information systems
 - 20 security and control families
 - E.g., Access Control, Awareness and Training, Monitoring
- Low-Impact Security Baseline: 131 controls
- Medium-Impact Security Baseline: 177 controls
- High-Impact Security Baseline: 188 controls
- Latest controls database publicly available¹

¹ <https://csrc.nist.gov/Projects/risk-management/sp800-53-controls/release-search/>

© 2024 UZH, CSG@ifi

14

ifi

ISO 27001

- General guidelines and certification
- Better for operationally mature organizations
 - Certification process: 6-18 months, Cost: from \$ 5,400 to \$ 20,000+
 - More recognized throughout the industry
 - Less technical, emphasizing risk-based management and best security practices

ISO 27001 certification comprises of 114 controls in 14 groups and 35 control objectives ensuring all the information covering people process supplier vendors and technology are safe and secure

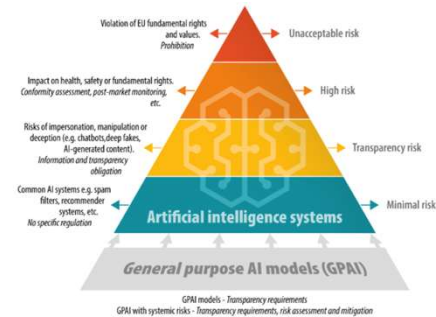


© 2024 UZH, CSG@ifi

15

ifi

EU AI Act



European Parliament (2024). Artificial intelligence act. [https://www.europarl.europa.eu/RegData/etudes/BRIE/2024/1688732/EPRS_BRI\(2024\)698732_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2024/1688732/EPRS_BRI(2024)698732_EN.pdf)

© 2024 UZH, CSG@ifi

16

ifi

Risk Definition

- Different risk management frameworks exist
 - Each one defining their own approach, with different steps
 - E.g., ISO 27005, ENISA ISMS, and NIST 800-30
- However, risk can be “viewed” as a single formula:

$$Risk = Impact \times Likelihood$$

- Where,
 - Impact = “Size” of cybersecurity incident result (both quantitative and qualitative)
 - Likelihood = How likely is the event to happen based on the company's profile

© 2024 UZH, CSG@ifi

ifi

© 2024 UZH, CSG@ifi

18

ifi

Processes for Risk Management

- Ongoing process of **assessing, treating, and monitoring risks**
 - Different types of risk can be considered, depending on the context
 - E.g., business, economic, and safety risks are possible



- Risk assessment is the necessary process of **identifying, analyzing, and evaluating risks** from different perspectives
 - E.g., technical, economic, legal, and societal

© 2024 UZH, CSG@ifi

19

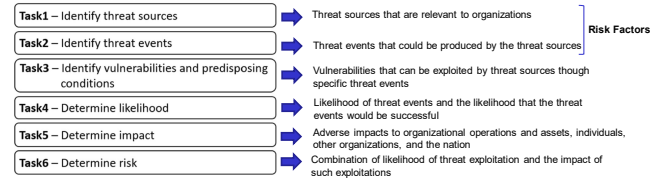
ifi

19

NIST 800-30 Overview

- **Goal:** Conduct NIST risk assessments in accordance with the NIST CSF recommendations
 - Translates cyber risk to the board members and CEOs
- One of the most **complex and difficult** to execute

Tasks for conducting Risk Assessments (NIST Special Publication 800-30)



© 2024 UZH, CSG@ifi

20

ifi

20

NIST 800-30: Risk Matrix

- NIST **recommends** to consider (a) threat source motivation and capability, (b) nature of the vulnerability, and (c) existence and effectiveness of current controls

The threat-source is highly motivated and sufficiently capable, and controls to prevent the vulnerability from being exercised are ineffective.

Exploitation of the vulnerability (i) may result in the highly costly loss of major tangible assets or resources, (ii) may significantly violate, harm, or impede an organization's mission, reputation, or interest; or (iii) may result in human death or serious injury.

Threat Likelihood	Impact		
	Low (10)	Medium (50)	High (100)
High (1.0)	10 X 1.0 = 10	50 X 1.0 = 50	100 X 1.0 = 100
Medium (0.5)	10 X 0.5 = 5	50 X 0.5 = 25	100 X 0.5 = 50
Low (0.1)	10 X 0.1 = 1	50 X 0.1 = 5	100 X 0.1 = 10

Impact > 50 = **High**
 10 < Impact < 50 = **Medium**
 Impact < 10 = **Low**

© 2024 UZH, CSG@ifi

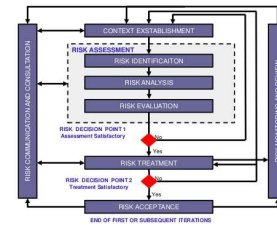
21

ifi

21

ISO 27005

- A core part of the ISO 27000 series
- Provides **more general view on the processes** involved in risk management
 - Does not provide recommendations, less technical, more about processes



© 2024 UZH, CSG@ifi

22

ifi

22

Comparison of Cybersecurity Frameworks

ISMS: Information Security Management System

Framework	Type	Key Elements	Goal	Main Stakeholders
NIST CSR	General Guideline	Identify, Protect, Detect, Respond, and Recover	Improve security of critical infrastructure	Governments and Companies
ENISA ISMS	General Guideline	Security Policy, Scope, Risk Assessment, Selection of Controls, Application	Implement appropriate measurements to eliminate or minimize the impact of security threats	Governments and Companies
ISO 27001	General Guideline / Certification	Organizational context, Scope, Leadership, Planning, Support, Operations, Performance evaluation, Improvement	Define requirements for the establishment, implementation, and maintenance of ISMS	Governments and Companies
ISO 27002	Security Controls and Implementation Guidance	Four category of controls: Organizational, People, Physical, Technological	Detail the needed security controls from an implementation perspective	Specialists
NIST SP 800-53	Security Controls and Implementation Guidance	Low, Medium, and High-Impact Security baseline, different controls	Define the processes and controls needed to comply with certain security certifications	Specialists
NIST SP 800-30	Risk Assessment	Identify threat sources, events, and vulnerabilities, Determine likelihood, impact, and risk	Conduct risk assessments in accordance with the NIST CSF recommendations	Specialists, Board members and CEOs
ISO 27005	Risk Management	Context establishment, Risk identification, analysis, and evaluation, Risk treatment and acceptance, Risk monitoring, communication, and review	Provide a general view on the processes involved in risk management	Governments and Companies
ENISA RM/RA	Risk Management	Same as of ISO 27005, but with definitions for long-, middle-, and short-term activities	Provide an overview of relevant content about risk management	Governments and Companies

© 2024 UZH, CSG@ifi

23

ifi

23

Part 2: Cybersecurity Economics

© 2024 UZH, CSG@ifi

ifi

24

Cybersecurity and Economics

- Cybersecurity economics as we know today has its start at the beginning of the 21st century
 - Many problems can be explained using microeconomic concepts
 - Network externalities and asymmetric information
 - Information security is a much deeper and more political problem than usually realized
- The higher the investment, the more secure the system?
 - Benefits are increasing at a decreasing rate
 - It is important to understand the risks and the associated cybersecurity costs

Why Information Security is Hard
– An Economic Perspective

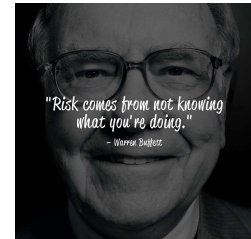
© 2024 UZH, CSG@ifi

25

ifi

25

Is that all about money?



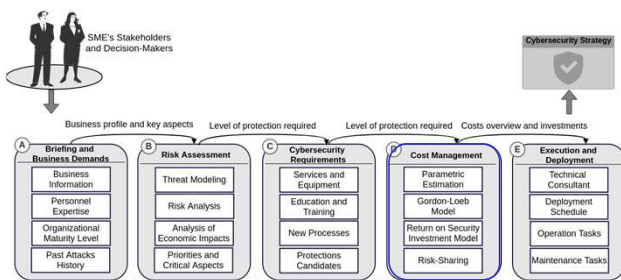
© 2024 UZH, CSG@ifi

26

ifi

26

Cybersecurity Planning for Investments



M. Franco: CyberTEA: a Technical and Economic Approach for Cybersecurity Planning and Investment; PhD Thesis, Communication Systems Group, University of Zurich, February 2023, Zurich, Switzerland

© 2024 UZH, CSG@ifi

27

ifi

27

Life-Cycle of Investments

Distributed Denial-of-Service (DDoS)

- Investing in cybersecurity is not a trivial task
 - Myriad of **threats**
 - DDoS Attacks, Zero-Day Vulnerability Exploits, Ransoms, and Crypto Jacking
 - Several **factors**
 - “Accept risk?” “Where to invest, on training or solution?” “How much?”
 - Wide range of **solutions**
 - Third-party vendors, closed/open source solutions, or cyber insurance
- The life-cycle of investing in cybersecurity as three steps:
 - Identifying threats, analysing risks, and implementing protections

© 2024 UZH, CSG@ifi

28

ifi

28

Identifying Threats

Machine Learning (ML)
Artificial Intelligence (AI)

- Penetration Testing (seen in previous classes)
 - Continuous and **proactive** approach
- Monitoring tools
 - Human operator in the loop → Difficult to automate the process
 - Research on ML/AI techniques
 - Networking context → Big data problem
- **Post mortem** analysis
 - Cybersecurity incident already happened → **reactive** approach
 - Information sharing incentives

© 2024 UZH, CSG@ifi

29

ifi

29

Determining the Economic Impact

Service Level Agreements (SLAs)

- If a cybersecurity incident occurs, what is a potential loss?
 - Direct financial losses
 - Direct effect on company's income (*i.e.*, lower profit and high cost)
 - Market-based
 - Not meeting SLAs → Loss of **market shares**
 - Reputational
 - Image of the company is damaged → Loss of customer **confidence**
 - Competitive Advantage
 - **Customer migration**, difficulty to **recover** from incident, **residual damage**
- Difficult to **quantify** → Need for a mature **methodology**

© 2024 UZH, CSG@ifi

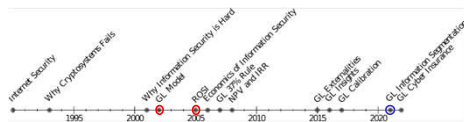
30

ifi

30

Models for Cybersecurity Economics (1)

- Models can be used as **benchmarks** to support cybersecurity decisions
 - E.g., **Gordon-Loeb (GL)** for the definition of the optimal budget and **Return on Security Investment (ROSI)** for the selection of cost-efficient solutions
- Important to assess, whether a **security strategy** is effective and worthwhile to receive an **investment**



© 2024 UZH, CSG@ifi

31

ifi

31

Models for Cybersecurity Economics (2)

- Most well-known and accepted models are the GL and ROSI
- GL** focuses on determining the **maximum amount of money** that should be invested to protect data or systems
 - Only a small fraction of the expected loss should be invested
- ROSI** focuses on determining if a specific solution is **cost-effective**
 - The risk mitigation effect should be greater than the expected costs

© 2024 UZH, CSG@ifi

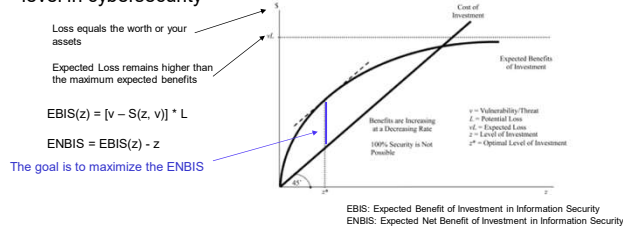
32

ifi

32

GL Model: Optimal Investment

- Economic model to analyze which is the **optimal investment level** in cybersecurity



L. Gordon, M. Loeb, L. Zhou: The Economics of Information Security Investment; ACM Transactions on Information and System Security, Vol. 5, No. 4, November 2002, pp. 1-21.

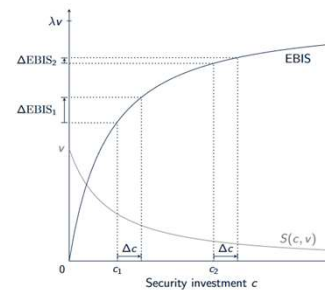
© 2024 UZH, CSG@ifi

33

ifi

33

GL Model: Optimal Investment

Moore, T. Security Metrics, Security Investment Models and Intro to R. Available: <https://tylemoores.ms.utulsa.edu/courses/econsec/slides/ra/theo2-handout.pdf>

© 2024 UZH, CSG@ifi

ifi

34

GL Model: Maximum Investment

- How much the data/system is **valued** (λ) ?
- How much the data/system is at **risk** (t) ?
- Probability** of an attack on the data/system such that it is going to be **successful** (v) ?
- The potential loss (d) is given by $d = \lambda * t * v$
- The Gordon-Loeb determined that the investment in security should not exceed 37% of the potential loss
$$\text{Investment} = d * 0.37$$

L. Gordon, M. Loeb, L. Zhou: The Economics of Information Security Investment; ACM Transactions on Information and System Security, Vol. 5, No. 4, November 2002, pp. 1-21.

L. Gordon, M. Loeb, L. Zhou: Investing in Cybersecurity: Insights from the Gordon-Loeb Model; Journal of Information Security, Vol. 7, No. 2, January 2016, pp. 49-59.

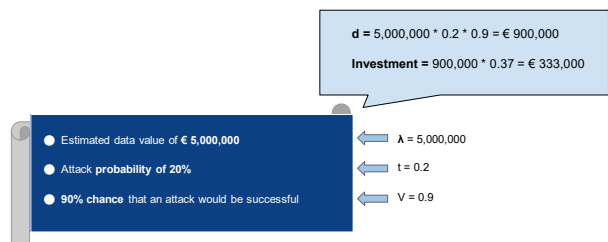
© 2024 UZH, CSG@ifi

35

ifi

35

Gordon-Loeb: Example



© 2024 UZH, CSG@ifi

36

ifi

36

GL Model: Information Segmentation

- Segments cover systems and information with specific values and interests for both company and attackers
 - Investments in cybersecurity should be done **per segment** since different requirements and information are needed
- To find the **optimal investment** per segment, four steps have to be followed:
 - Step 1: Estimate the value and the potential loss (L) of each segment
 - Step 2: Estimate the probability (v) of each segment's information being victim of a successful cyberattack
 - Step 3: Create a grid with all possible combinations of Step 1 and Step 2
 - Step 4: Derive the level of cybersecurity investment by increasing the investment as long as the benefit of the additional investment is greater than or equal to the cost of additional investment

L. Gordon, M. Loeb, L. Zhou: Information Segmentation and Investing in Cybersecurity. *Journal of Information Security*, Vol. 12, January 2021.

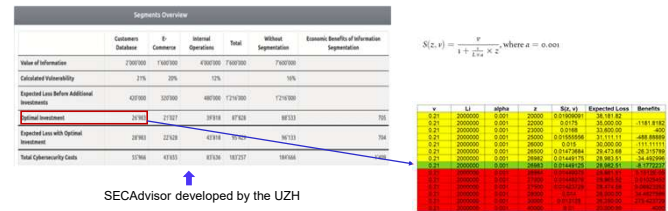
© 2024 UZH, CSG@ifi

37

ifi

37

GL Model: Example of Optimal Investment



L. Gordon, M. Loeb, L. Zhou: Information Segmentation and Investing in Cybersecurity. *Journal of Information Security*, Vol. 12, January 2021.

© 2024 UZH, CSG@ifi

38

ifi

38

Return On Security Investment (ROSI)

- Support cybersecurity decisions by doing cost-benefit analysis of protections
 - How can you support a decision for a **specific antivirus** with different prices, characteristics, and mitigation rates?
- ROSI is one of the **most used metrics** among corporate information security managers
 - It is slightly similar to the Return on Investment (ROI) but focuses on **reducing losses** instead of maximizing profits

$$ROSI = \frac{\text{Expected Returns} - \text{Cost of the investment}}{\text{Cost of the investment}}$$

$$ROSI = \frac{(ALE \cdot \text{mitigation rate}) - \text{Cost of the investment}}{\text{Cost of the investment}}$$

ALE: Annual Loss Expectancy

39

ENISA: Introduction to Return on Security Investment, 2012. <https://www.enisa.europa.eu/publications/introduction-to-return-on-security-investment>

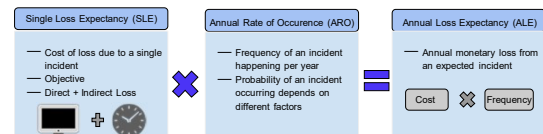
© 2024 UZH, CSG@ifi

39

ifi

39

Annual Loss Expectancy (ALE)



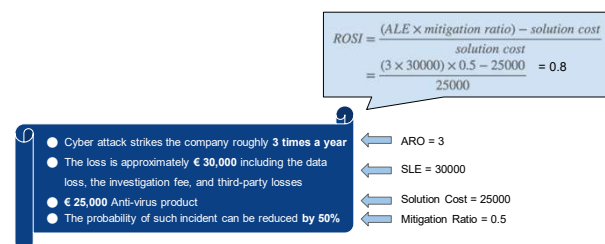
© 2024 UZH, CSG@ifi

40

ifi

40

ROSI Example



© 2024 UZH, CSG@ifi

41

ifi

41

Summary of Cybersecurity Economics

- Cybersecurity economics involves a broad range of activities
 - Education, prevention, monitoring, maintenance, remediation, insurance
 - GL and ROSI are the most accepted models
- It is critical to **map systems and processes** for an efficient risk management, which guides the cybersecurity investment strategy
 - Frameworks, standardizations, and techniques
 - Training and education** focusing on the demands and threats of the different sectors
- Cybersecurity is **fundamental** to avoid losses for companies
 - Investment should be done on a per-segment-basis

© 2024 UZH, CSG@ifi

42

ifi

42

Key Takeaways

- ❑ Risk Management is an important pillar for cybersecurity planning
 - Identification, evaluation, and mitigation of threats
 - A efficient risk assessment is critical for an efficient cybersecurity strategy
- ❑ Several frameworks and certifications available, different complexities
 - SMEs still struggling with them
 - Costs vs Experts vs Awareness
- ❑ Cybersecurity economics provides models and benchmarks for decision-making
 - There is no one-size-fits-all approach
- ❑ Roadmap of cybersecurity shows a long path for the next years
 - Short-term: Education and Training
 - Medium-term: Efficient risk assessment, standards and regulations
 - Long-term: Better strategies and wider adoption of cybersecurity

Computer Network and Security Principles (CNSP)

L10 – Cybersecurity and Machine Learning (ML)

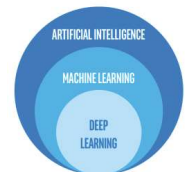
Cybersecurity and Machine Learning (ML)

- Objectives
 - To understand the basics of ML and how it can help network cybersecurity
 - To understand the basics of Federated Learning (FL) in the network cybersecurity scenario
- Topics
 - Supervised and unsupervised ML fundamentals and their application in network cybersecurity
 - Challenges of ML in network cybersecurity
 - Horizontal and vertical FL applied to network cybersecurity
 - Challenges of FL in network cybersecurity
 - Summary of recommendations and takeaways

ML FUNDAMENTALS

ML Definition

- Machine Learning (ML) is a field, part of artificial intelligence, devoted to building systems that learn
 - Systems that use data to generalize past experiences in order to predict future outcomes
- Statistical analysis is a core part of machine learning
- Examples of ML in Cybersecurity
 - Spam detection
 - Malware detection
 - Access control
 - Users' authentication



ML Fundamentals

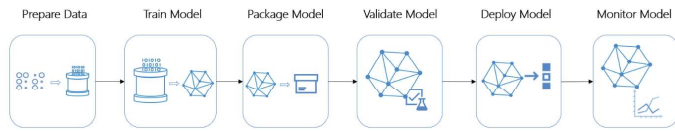
- Recently, ML has gained prominence due to
 - Increase in the data collected in almost all environments
 - Increase in the processing power (CPUs, GPUs)
 - Research in new techniques and algorithms
- ML has the capacity to detect complex data patterns, handle multi-dimensional, and multi-variate data
 - ✓ – Avoid expert knowledge to define policies (time consuming)
 - ✓ – Adapt themselves to dynamic scenarios
 - ✓ – Predictions are usually hard to explain (black-box)
 - ✓ – Need large amount of data and resources for training

ML Terminology

- Feature: input variable (X). A machine learning project might use a single feature or millions of them
- Sample: vector of features with particular values
- Label: ground true of a sample (Y)
- Dataset: set of samples with or without labels (supervised vs unsupervised)
- Algorithm: method by which the AI system conducts its task, generally predicting output values from given input data.
- Model: result of training an algorithm with a dataset

ML Pipeline

- Although data is now more than available than ever, ML/DL algorithm usage is not always direct
- Typically, an ML/DL **pipeline** is followed to create and evaluate models



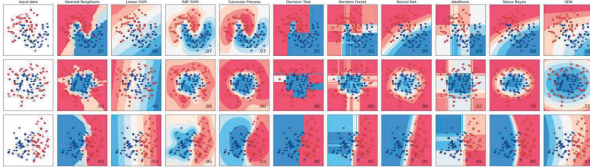
Source: <https://towardsdatascience.com/i-had-no-idea-how-to-build-a-machine-learning-pipeline-but-heres-what-i-figured-f3a7773513a>

ML Pipeline: Prepare Data

- Remove **erroneous** samples and features
- Remove **highly correlated** features
- Data **normalization and feature extraction**
 - Statistical feature extraction: Average, Max, Min, Std. Dev.
 - Time series: Fourier transform, Dynamic Time Warping
- Data **balancing/augmentation**
- Dataset **division**
 - 60-80% for model training
 - 10-20% for model validation (hyperparameter selection)
 - 10-20% for model testing

ML Pipeline: Train Model

- **Training dataset** is used for **model generation**
- **Validation dataset** is used for **hyperparameter** selection and optimization
 - Cross-validation → Iteratively generate train / validation datasets
- **Testing dataset** is used for **performance** evaluation
- Different algorithms should be tested to select the best one



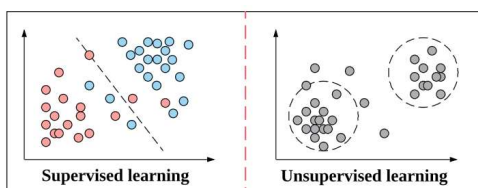
Source: https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

ML Pipeline: Remaining Steps

- **Package model** to use it everywhere
 - Generated scalers or feature extractors should be included in the package
- **Validate model** using test dataset (data not seen for training or hyperparameter selection)
- **Deploy model** in the production or real environment where it will be applied to make predictions or process data
- **Monitor model** is needed to ensure its correct functioning in time

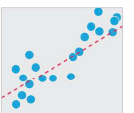
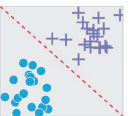
ML Basics

- ML/DL algorithms can be classified in two main categories according to **data morphology**:
 - **Supervised**, when data is labelled
 - **Unsupervised**, when data is unlabelled



Supervised Learning

- The goal is to create models able to predict the output of **samples** based on **training labelled data**
- **Classification** algorithms: Predict the **class** to which unseen data vectors belong to
 - Binary classification: label data as normal/abnormal
 - Multi-label classification: One sample belongs to many labels at the same time
- **Regression** algorithms: Predict the **output values (continuous)** based on the data fed to the system



Classification

□ ML classification algorithms

- Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM), k-Nearest Neighbours
- Ensemble methods: Combine many classifiers to output a common decision

□ DL classification algorithms

- Multi-Layer Perceptron
- Long Short-Term Memory (LSTM) for time series
- Convolutional Neural Network (CNN): 2D for images

□ In cybersecurity, classification algorithms are used for

- Classification of cyberattacks or malware families
- Classification of malicious behaviour

Regression

□ ML regression algorithms

- Linear Regression (LR), Polynomial Regression (PR), Random Forest Regressor (RFR), etc

□ DL regression algorithms

- Multi-Layer Perceptron (the same as for classification, only predicted output changes)
- LSTM for time series
- CNN: 2D for images, 1D for time series

□ In cybersecurity, regression algorithms are used for

- Predict Risks
- Predict the cost of cyberattacks

Unsupervised

□ Data vectors are not labelled, samples only contain input data (features, X)

- Used to extract patterns according to probability densities on the data

□ Three main applications of unsupervised learning are

- Dimensionality Reduction: Reduce the number of variables or features
- Clustering: Group samples according to their similarities
- Anomaly Detection: Identify abnormal samples, events, or observations

Dimensionality Reduction

□ The goal is to reduce the number of features under consideration without losing meaningful information

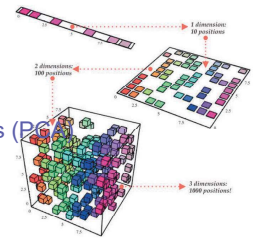
- Speed up computation
- Reduce overfit and noise
- Reduce resource consumption

□ Algorithms

- ML → Principal Component Analysis (PCA)
- DL → Encoder (half Autoencoder)

□ Applied in cybersecurity for

- Pre-process data
- Detect Anomalies (combined with statistical algorithms such as Euclidean Distance)



Clustering

□ The goal is to group samples according to their similarities

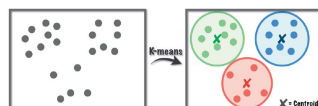
- Can be used to label data (infer data classes)

□ Algorithms

- ML → K-means, DBSCAN, Gaussian Mixture Models
- DL → Autoencoder + Cluster hardening loss

□ Applied in cybersecurity for

- Inferring cyberattacks or malware types (1 cluster per malware)
- Anomaly detection (2 clusters)



Anomaly Detection

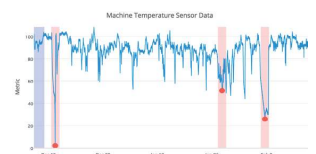
□ The goal is to identify rare samples based on a set of unlabelled data points and the assumption that most of the training data is normal

□ Algorithms

- ML → OC-SVM, LOF, Isolation Forest,
- DL → Autoencoder (MLP, CNN, RNN,...) + Reconstruction error (MSE, MAE,...)

□ Applied in cybersecurity for

- Detect Zero-day attacks



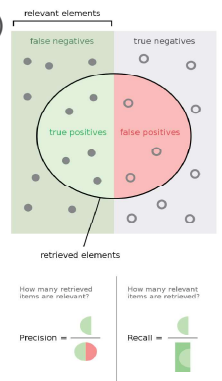
ML in Network Cybersecurity

- **Classification of DDoS Attacks**
 - Data → PCAP files
 - Number of packets, packet length, packet type, bytes sent and received, etc
 - Model → Classification
 - Random Forest (RF), k-Nearest Neighbour (k-NN)
- **Detection of Zero-day Ransomware spreading phase**
 - Data → Netflow files
 - Statistics of flows (duration, bits, packets, etc)
 - Model → Anomaly Detection
 - Isolation Forest (IF), One Class-Support Vector Machine (OC-SVM), Local Outlier Factor (LOF), etc

Typical ML Metrics for Cybersecurity

□ Detection of Anomalies (unsupervised)

- True Positive Rate (TPR)
- False Positive Rate (FPR)
- True Negative Rate (TNR)
- False Negative Rate (FNR)



□ Classification (supervised)

- Precision
 - Recall
 - F1-Score
- $$\text{Precision} = \frac{tp}{tp + fp}$$
- $$\text{Recall} = \frac{tp}{tp + fn}$$
- $$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2tp}{2tp + fp + fn}$$

ML Frameworks, Libraries, and APIs

- **Scikit learn**: tool that implements many Machine Learning algorithms in python
- **TensorFlow**: library for distributed numerical computation. It makes it possible to train and run very large neural networks efficiently
- **Keras**: high-level Deep Learning API that makes it very simple to train and run neural networks.

Challenges of ML in Cybersecurity

- **Bad data** (not proper labels, not representative data)
- **Data privacy** (in scenarios with sensitive data)
- **Data bias** (not representative or enough data)
- **Computational, storage and memory consumption** to train models
- **Bad algorithm selection** or ML pipeline implementation
- **Overfitting and underfitting**
- **Untrusted models and predictions**

Data Bias

- **Insufficient Quantity of Training Data** → Models need a certain quantity of samples (depends on the model complexity)
 - **Solution**: Data augmentation, synthetic data
- **Non-representative or poor-quality training data** → Testing data differs from training data
 - **Solution**: Obtain better data for training
- **Irrelevant features** → Features highly correlated, constants, or random
 - **Solution**: Perform feature engineering process to extract and select good features

Overfitting

- **Overfitting**
 - Good performance for training, bad for testing
 - The model is too complex for the amount and noisiness of the training data
- **Solutions**
 - Simplify the model by selecting one with fewer parameters
 - Constrain the model parameters
 - Reduce the number of features in the training data
 - Gather more training data
 - Reduce the noise in the training data

Underfitting

Underfitting

- Bad performance for training and testing
- The model is too simple to learn the underlying structure of the data

Solutions

- Select a more powerful model with more parameters
- Feed better features to the learning algorithm (feature engineering)
- Reduce the constraints on the model (e.g., reduce the regularization hyperparameter)

FL FUNDAMENTALS

FL Fundamentals

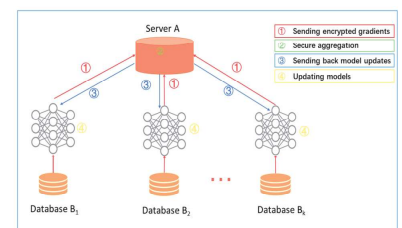
Federated Learning (FL) is a paradigm introduced by Google in 2016 to deal with the problems of

- Sharing **sensitive data** to train ML models
 - Public awareness and regulations on data privacy
- Maintaining **data in central silos**
 - Real-world data is often decentralized across many parties
- Sending **data** may be **too costly**
 - Devices with limited bandwidth/power



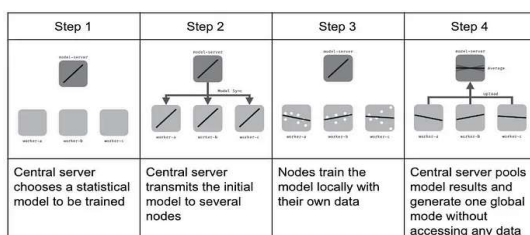
FL is an ML paradigm where a collaborative global model is trained in a privacy-preserving way by aggregating models that have been trained locally by clients.

- Clients can be very **large in number** and face challenges of **data and network heterogeneity**



FL Lifecycle

FL trains an ML model between the participants of a federation without sharing sensitive data



FL Lifecycle

1. One participant (usually the aggregation server) **decides the model used by all clients**
2. Each **participant trains a local model** only with its own **private data**
3. Each **participant shares the weights** of its local model to the **aggregator**
4. **Weights are aggregated** in a central (o distributed) manner to **create a global model**
5. The **global model is downloaded** by each client
6. Steps 1 - 4 are repeated for several rounds until the model converges

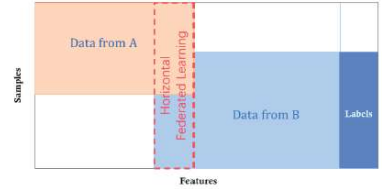
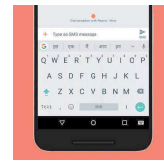
FL Scenarios

X: features
Y: labels
I: IDs of participants
D: local datasets

- FL presents **three scenarios** according to the **characteristics of the participants data**
 - Horizontal Federated Learning (HFL)**
 - Participants share the same set of features but different samples
 - $X_i = X_j, Y_i = Y_j, I_i \neq I_j, \forall D_i, D_j, i \neq j$
 - Vertical Federated Learning (VFL)**
 - Participants share the same samples or sample ID but different set of features
 - $X_i \neq X_j, Y_i = Y_j, I_i = I_j, \forall D_i, D_j, i \neq j$
 - Federated Transfer Learning (FTL)**
 - Participants only share a reduced subset of samples ID and features
 - $X_i \neq X_j, Y_i \neq Y_j, I_i = I_j, \forall D_i, D_j, i \neq j$

Horizontal Federated Learning

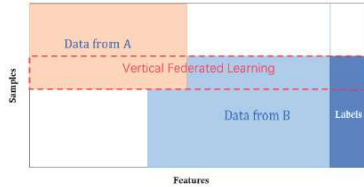
- Horizontal Federated Learning (HFL)
 - Participants share the **same** set of **features** but **different samples**
 - $X_i = X_j, Y_i = Y_j, I_i \neq I_j, \forall D_i, D_j, i \neq j$
- Use case: Learning users' keyboard behaviors and word selection



Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-19.

Vertical Federated Learning

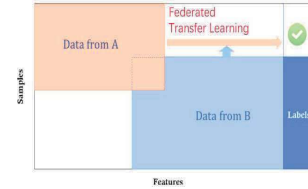
- Vertical Federated Learning (VFL)
 - Participants share the **same samples** or sample ID but **different set of features**
 - $X_i \neq X_j, Y_i = Y_j, I_i = I_j, \forall D_i, D_j, i \neq j$
- Use case: Different Organizations collaborating to improve their services



Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-19.

Federated Transfer Learning

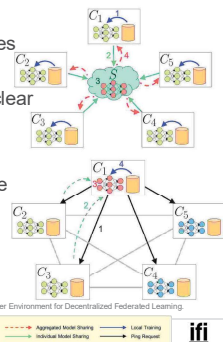
- Federated Transfer Learning (FTL)
 - Participants share only a **subset of samples** and **features**
 - $X_i \neq X_j, Y_i \neq Y_j, I_i = I_j, \forall D_i, D_j, i \neq j$
- Use Case: Two different organizations with a subset of common samples and features



Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-19.

FL Architectures

- The **training protocol** can be **implemented** following **two approaches**:
 - Centralized**
 - It is the most used nowadays because simplifies the aggregation task
 - The aggregation server is a bottleneck and a clear target for attackers
 - Distributed**
 - It is more robust against some attacks because the aggregation task is distributed
 - It adds complexity to the aggregation task because participants have to collaborate



Guha Roy, Abhijit & Siddiqua, Shayan & Pöndel, Sebastian & Nawab, Nassir & Wachinger, Christian. (2019). BrainTorch: A Peer-to-Peer Environment for Decentralized Federated Learning

FL Aggregation Functions

- The weights of each participant can be aggregated according to different aggregation functions
 - Federated Averaging (FedAvg)**: it calculates the mean average of the local models weights
 - Trimmed mean**: It extends FedAvg by excluding the highest and the lowest weight updates when calculating the mean of the local models weights
 - Different configurations of this function can be applied to exclude from 1 to n weights
 - Coordinate wise**: It uses the median of every model weight instead of the mean average

FL in Network Cybersecurity

- Collaborative and privacy-preserving Anomaly Detection and classifiers
 - Detect anomalies produced by cyberattacks affecting one or more devices of the federation
 - Classify malware in a collaborative way
 - Reduce or avoid the training phase of new devices joining a federation
 - Create more robust models containing the behaviors of several heterogeneous devices

ML Frameworks and Libraries

- TensorFlow Federated (TFF) is an open-source framework for ML and other computations on decentralized data
- PySyft is an open-source multi-language library enabling secure and private machine learning by wrapping and extending popular deep learning frameworks such as PyTorch
- Flower is a unified approach to federated learning, analytics, and evaluation. Federate any workload, any ML framework, and any programming language

Typical FL Metrics for Cybersecurity

- Detection of Anomalies (unsupervised)
 - TPR, FPR, TNR, FNR
- Classification (supervised)
 - Precision
 - Recall
 - F1-Score
- Apart from the previous ones, there are others such as
 - Network overhead
 - Training time
 - Robustness against adversarial attacks

FL Challenges

- Data privacy vs. performance of centralized solutions with no privacy-preserving capabilities
- Training time while providing good performance
- Communications overhead during training
- Clients trustworthiness
- Performance with non-IID (non-independent and identically distributed) data
- Adversarial attacks destroying the model performance or inferring sensitive data

SUMMARY

Summary (1)

- ML has the capacity to detect complex data patterns
- Malware detection, Access control, or authentication are examples of ML applied to cybersecurity
- Classification (supervised) and anomaly detection (unsupervised) are the most used approaches in cybersecurity
- Precision, Recall and F1-Score are the metrics more used with supervised models
- Scikit learn, TensorFlow, and Keras are the most well-known frameworks, libraries and APIs

Summary (2)

- ❑ FL pretends to solve the traditional ML problem of data privacy and central silos
- ❑ Collaborative anomaly detection and malware classification are examples of FL in cybersecurity
- ❑ Horizontal FL is currently the most explored scenario in the cybersecurity field
- ❑ FedAvg is the most used aggregation function but others such as Trimmed mean provide robustness
- ❑ Network overhead and training time are also relevant metrics for FL applied to cybersecurity

Computer Network and Security Principles (CNSP)

L10 – Concerns of Machine and Federated Learning (ML/FL)

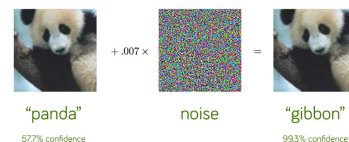
Concerns of Machine and Federated Learning

- Objectives
 - To overview the concerns and issues of ML and FL in terms of cyber attacks
 - To overview societal implications of ML
- Topics
 - Main adversarial attacks and countermeasures affecting ML
 - Main adversarial attacks and countermeasures in FL
 - Pillars and metrics to trust ML predictions
 - Pillars, notions, and metrics to trust FL predictions
 - Summary of recommendations and takeaways

ADVERSARIAL ATTACKS AFFECTING ML

Definition of Adversarial ML

- Adversarial ML is a method that aims to trick machine learning models by providing deceptive inputs
 - Samples can be manipulated during training or/and testing to launch adversarial attacks with different purposes
 - The most extensive studies have been conducted in computer vision, but network security is also a target scenario



Terminology

- An adversarial attack is a method that generates adversarial samples to affect the model predictions
- An adversarial sample is a corrupted version of a valid sample, where the modification is done perturbing:
 - One or more features (supervised and unsupervised ML)
 - The label (supervised ML)



Attacker's Knowledge

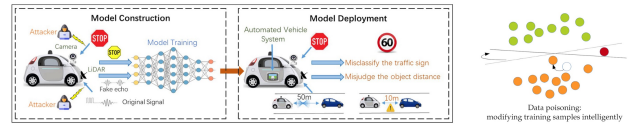
- Perfect-knowledge: correspond to white-box attacks, where the attacker has complete access to:
 - Training and testing datasets
 - Model details (algorithm and parameters)
 - Model predictions
- Limited knowledge: correspond to gray-box attacks, where the attacker partially knows the previous three aspects
- Zero-knowledge: correspond to black-box attacks, where the attacker has only access to:
 - Model predictions

Types of Adversarial ML

- Poisoning Attacks
 - The attacker **modifies training samples and/or their labels** with the **aim of corrupting the learning model**
 - Requires **access to the training dataset**
- Evasion Attacks
 - The attacker **manipulates testing samples to affect their correct classification** (relevant in malware detection)
 - **Most prevalent and realistic** attack type
- Inference Attacks
 - The attacker **targets grey and black-box ML systems to either reconstruct the model or extract training data** (relevant when training data and/or the model parameters are confidential)

Poisoning Attacks

- Inject **fake samples during training** with the aim of **corrupting the learned model**
 - Ways to implement poisoning attacks: **label modification, data injection, data modification, or backdoors**

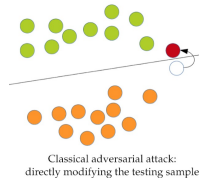


- **Cybersecurity example:** flip the labels of malicious and benign samples to destroy malware detection systems

Evasion Attacks

- **Modify samples** that are going to be **evaluated during testing**
 - These attacks are the **most practical and used** on intrusion detection and malware scenarios

- **Cybersecurity example:**
Modify samples modelling malware behavior to evade its detection



Classical adversarial attack: directly modifying the testing sample

Existing Evasion Attacks

- Some of the most well-known and used adversarial attacks are:
 - FSGM (Fast Gradient Sign Method)
 - L-BFGS (Limited-memory FGSM)
 - JSMA (Jacobian-based Saliency Map Attack)
 - Deepfool Attack
 - C&W (Carlini & Wagner)
 - GAN (Generative Adversarial Networks)
 - ZOO (Zeroth-order Optimization)

FGSM and JSMA Attacks

- FastGradient Sign method (FGSM) **minimize the amount of perturbation** added to any pixel of an image to cause misclassification
 - ✓ – Efficient computing time
 - ✓ – Perturbations are added to **every** feature (not useful for network security with categorical features)
- Jacobian-based Saliency Map Attack (JSMA) uses **feature selection** to minimize the number of features modified while causing misclassification.
 - ✓ – Very **few features** are perturbed
 - ✓ – More **computationally intensive** than FGSM

Deepfool Attack

- Deepfool Attack generates **untargeted adversarial samples** by **minimizing the Euclidean distance** between perturbed samples and original samples
 - ✓ – Effective at producing adversarial examples, with fewer perturbations and higher misclassification rates
 - ✓ – More computationally intensive than FGSM and JSMA
 - ✓ – Adversarial examples are likely not optimal

GAN Adversarial Attack

- Generative Adversarial Networks (GAN) consists of two neural networks competing each other
 - One network acts as a **generator**, and the another one behaves as the **discriminator**
 - The generator tries to produce samples that the discriminator will misclassify
 - The discriminator tries to distinguish real samples from ones created by the generator
- ✓ – Training a Generate Adversarial Network is **very computationally intensive** and can be **highly unstable**

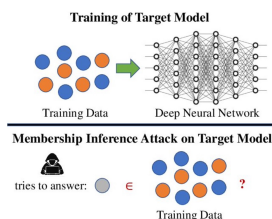
Countermeasures for Evasion Attacks

- Adversarial training
 - Consists of using adversarial attacks during training to build robust models against those attacks
 - Each attack implementation needs a independent training phase
- Spam filter example:
 - Evasion attacks try to obfuscate spam messages through the misspelling of "bad" words.
 - Adversarial Training could be to integrate these misspelled or bad words during training with the label of Spam

Inference Attacks

- These attacks are executed in **black- and grey-boxes approaches during testing** to:

- Learn the **labels or classes** used to **train the model**
- Learn the **training data**
- Learn the **model and its parameters**
- Learn the **predictions** done by the model



ADVERSARIAL ATTACKS AFFECTING FL

FL Security Threats

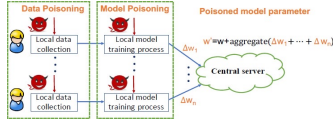
- Federated Learning (FL) models are particularly **vulnerable to poisoning attacks**
 - Due to the number of clients collaborating to train models
- Attacks can be launched from different points
 - **Inside**: one or more participants of the federation
 - **Participants** and the **aggregation server** can launch attacks
 - **Outside**: third parties eavesdropping the communication between the federation participants
- **Insider attacks** are generally **more damaging** because they have more control over the FL architecture

FL Security Threats

- Attacks executed **during training** influence and corrupt the FL model
 - Data poisoning
 - Model poisoning
 - Backdoors
- Attacks launched **during testing** produce **wrong** FL model **predictions** and **infer** the **model parameters** (do not corrupt the model)
 - White- and grey-box attacks
 - Gradient leakage attacks

Poisoning Attacks

- Data and Model poisoning attacks attempt to **modify the behavior of the target model** in some undesirable way
 - **Data Poisoning**: values of some features are modified to corrupt the global model
 - **Model Poisoning**: some weights or gradients are modified to corrupt the global model



Backdoor Attacks

- Backdoor attacks aim to **predict affected samples as a target label**
 - The **attacker** needs to have **access** to the **training and testing datasets**
 - This attack does not affect the model behavior on clean samples (not easily detectable)
- During training, a given **watermark** is associated with the **target label**
- During testing, a target sample is **modified** with the **watermark**

Countermeasures

- Poisoning Attacks
 - **Robust aggregation mechanisms** can be used to **filter extreme weights and gradients** shared by malicious clients of the federation
 - **Methods detecting anomalies** can be used to filter extreme weights and gradients
- Inference Attacks
 - **Differential Privacy** → consists of adding noise to the weights or gradients shared from clients to the aggregator server
 - **Noise** affects client weights but when they are combined with those from other clients the **performance is not heavily affected**

SOCIETAL IMPLICATIONS OF ML

ML Trustworthiness

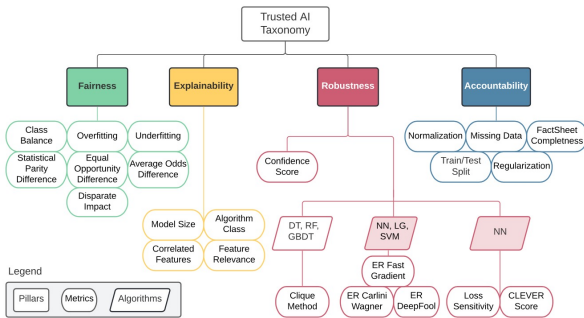
- Over the last decade, ML improved vital challenges of communication networks
 - Examples: detection of cyberattacks, real-time reconfiguration of network services, optimization of traffic routes, etc.
- It is critical to quantify the **trustworthiness level of ML models and their predictions**
 - The goal is to **calculate the trust score of ML models**



ML Trustworthiness Pillars

- Pillars relevant to measure the trustworthiness of ML
 - **Fairness**: focuses on creating ML models that provide unbiased and fair predictions
 - **Explainability**: enables a deeper understanding of the inner workings of ML models
 - **Robustness**: deals with the ability of ML models to deal with adversarial attacks
 - **Assurance or Accountability**: considers the quality of the training methodology, its accountability, and the transparency in which stakeholders are notified

Trusted Pillars and Metrics



Fairness

- **Metrics** to calculate the ML models **fairness**
 - **Underfitting**: detects if the model is unable to learn the relationship between inputs and outputs accurately
 - **Overfitting**: measures the model generalization capabilities
 - **Class Balance**: measures the ratio of samples belonging to different classes in the training dataset
 - **Statistical Parity Difference**: computes the spread between the percentage of samples belonging to the majority group receiving a favourable outcome compared to a protected group

Explainability

- **Metrics** to calculate the ML models **explainability**
 - **Algorithm Class**: The algorithm type is relevant to compute the model explainability
 - **Correlated Features**: high correlation among features might lead to biases in most explanation techniques
 - **Feature Relevance**: the lower irrelevant features, the better, as they would only make explanations more complex without being relevant
 - **Model Size**: indicates the number of parameters used by models

Robustness

- **Metrics** to calculate the ML models **robustness**
 - **Confidence Score**: measures the probability of predicting correctly samples
 - **Loss Sensitivity**: calculates the largest variation of the output of a Neural Network under a small change in its input
 - **Cross Lipschitz Extreme Value for Network Robustness (CLEVER) Score**: measures the minimal perturbation that is needed to change the classification outcome using the local Lipschitz constant
 - **Empirical Robustness**: measures the average minimal perturbation that needs to be introduced to change the model prediction

Accountability

- **Metrics** to calculate the ML models **accountability**
 - **Train/Test Split**: measures the ratio between the number of samples used for training and testing
 - **Missing Data**: evaluates how missing values of features of the training dataset are handled
 - **Normalization**: helps mitigating the effects of outliers and ensures that features values are in the same range
 - **Regularization**: avoids model parameters taking extreme values, and it is necessary to avoid memorization during training

SOCIETAL IMPLICATIONS OF FL

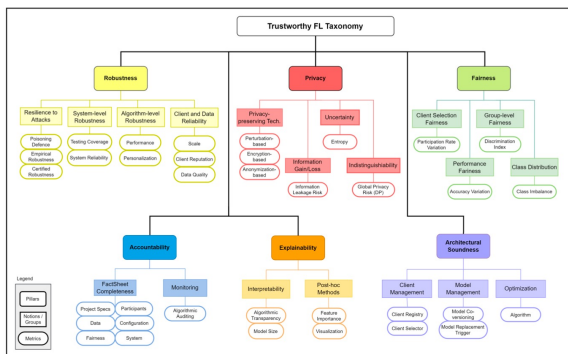
FL Trustworthiness

- FL is subject to **the same risks as ML**
 - Algorithmic bias, adversarial attacks, data privacy breaches, reliability issues, etc.
- In addition, FL involves more stakeholders, information exchanges, communication infrastructures and attack surfaces presenting **new challenges** such as:
 - Architectural design
 - Privacy-preserving standards
 - New perspectives on fairness and explainability

FL Trustworthiness Pillars

- Pillars relevant to measure the trustworthiness of ML
 - Robustness**: considers the resilience to attacks of the FL model and the **clients reliability**
 - Privacy**: quantifies how much sensitive data is preserved
 - Fairness**: calculates the fairness of models predictions and **federation participants selection**
 - Accountability**: considers the transparency of FL models and architectures
 - Explainability**: focuses on explaining the predictions of FL models
 - Architectural Soundness**: analyses architectural details of the federation

FL Trustworthiness



FL Robustness

- Robustness notions, mechanisms, and metrics
 - Resilience to Attacks**
 - Mechanisms to aggregate weights and detect outliers, metrics to measure the impact of model poisoning attacks, and approaches to set the minimum perturbation for attacks success
 - System-level Robustness**
 - Metrics to measure the robustness and reliability of systems considering errors in messages, clients, aggregations, etc
 - Algorithm-level Robustness**
 - Metrics to measure the algorithm performance and its personalization
 - Client and Data Reliability**
 - Metrics considering the number of clients, their reputation, and data quality to compute client reliability

FL Privacy

- Privacy notions, mechanisms, and metrics
 - Privacy-preserving Approaches**
 - Differential privacy, homomorphic encryption, and k-anonymity are some of the techniques used to improve data and weights privacy
 - Information Gain/Loss**
 - Aspects like the information leak risk of federation clients are important to quantify the data privacy offered by federated scenarios
 - Uncertainty**
 - High uncertainty in the adversary's estimation correlates with high privacy and Entropy is used to measure this aspect
 - Indistinguishability**
 - Privacy is high if the adversary cannot distinguish between two model's outcomes. Global Privacy risk computes this aspect

FL Fairness

- Fairness notions, mechanisms, and metrics
 - Client Selection Fairness**
 - Metrics considering clients participation rate variation are important to offer personalized rewards
 - Performance Fairness**
 - The variations in terms of accuracy or other performance metrics are also relevant to quantify the model fairness
 - Group-level Fairness**
 - The discrimination index is used to measures the difference in performance between a particular demographic group and the rest
 - Class Distribution**
 - Heterogeneity problem in class distribution can be closely related to client selection fairness. Class imbalance results in a slow converge rate of the global model

FL Accountability

- Accountability notions, mechanisms, and metrics
 - **Factsheet Completeness**
 - Project specifications, Participants information, data information, Federation configuration, models performance, and system information and important aspects taken into account to measure the accountability of FL models
 - **Monitoring**
 - SMACTR is an internal auditing framework for FL that includes five stages of scoping, mapping, artifact collection, testing and reflection

FL Explainability

- Explainability notions, mechanisms, and metrics
 - **Interpretability**
 - **Algorithm Transparency:** A model is considered transparent if it is understandable by itself. Interpretable models are: linear regression, logistic regression, decision trees, decision rules, k-nearest neighbors (KNN), Bayesian models
 - **Model Size:** The larger the model size the harder it is to understand the relationship between input and output
 - **Post-hoc Explainability Methods**
 - The importance of each feature is relevant to quantify the models explainability

FL Architectural Soundness

- Architectural Soundness notions, mechanisms, and metrics
 - **Client Management**
 - The **selection and registration of clients** are important tasks affecting the trustworthiness of FL models.
 - **Model Management**
 - Maintaining several **versions of FL models** and **replacing models** when the performance decreases are good practices to trust predictions of FL models
 - **Optimization**
 - Different **aggregation mechanisms** and **algorithms** not only optimize the FL models performance but also provide trust due to robustness and stability reasons

SUMMARY

Recommendations Adversarial Attacks

- Concerns in Machine learning
 - Poisoning, Evasion and Inference attacks affect ML models, being **Evasion attacks the most common and realistic ones**
 - **Adversarial training** is the most **useful countermeasure** against evasion attacks
 - Limitation of retraining models with adversarial samples per attack
- Concerns in Federated learning
 - **Poisoning** (data, model) and **Inference** are the **most common attacks** affecting FL models
 - Robust aggregation and differential privacy are the most **common countermeasures** against poisoning and inference attacks

Recommendations Trusted AI

- Trustworthy Machine learning
 - **Fairness, Explainability, Robustness, and Accountability** are the four main pillars quantifying the trustworthiness of ML predictions
- Trustworthy Federated learning
 - Apart from the four pillars of traditional ML, aspects dealing the **federation architecture, the client selection criteria, or the network topology** are important to quantify the trustworthiness of FL models

Computer Network and Security Principles (CNSP)

L12 – Selected Cybersecurity Topics

Selected Cybersecurity Topics

- Research at CSG
 - To overview **selected** cybersecurity issues and projects
 - Blockchain Signaling System (BloSS): Enabling a Cooperative and Multi-domain DDoS Defense
 - Framework to Detect Ransomware Affecting Linux-based and Resource-constrained Devices
 - Collaborative Threat Modeling
 - Smart Contract's Security
- Topics
 - Cooperative DDoS Defense
 - Guests: Getoar Gallopeni (PwC)

© 2024 UZH, CSG@UZH

1



1

© 2024 UZH, CSG@UZH

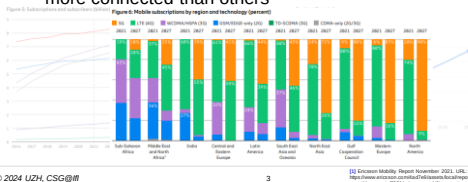
2



2

Preamble (1): Highly Connected Society

- Most of us live in an increasingly **connected society**
 - Ericsson Mobility Report November 2021 [1]
 - As of 2021, a bit more than **8 billion (bn)** **mobile subscriptions**
 - As of 2020, the world's population is estimated around 8 bn.
 - Naturally, such **distribution is not equal** and some regions are more connected than others



© 2024 UZH, CSG@UZH

3



3

5G Connectivity

As of November, 2021

As of November, 2024

© 2024 UZH, CSG@UZH

4



4

Preamble (2): Highly Connected Society

- Studies [2,3] show that the **degree of connectivity of a society is linked to its social development**
 - A higher degree of connectivity → higher social development
 - More connected devices, larger infrastructure, higher exchange of information
- On the one hand: 
 - To which extent highly connected societies are **resilient to a partial or total disruption** of such critical infrastructures?
- On the other hand: 
 - What are the **tools and strategies to disrupt critical infrastructures?** 

© 2024 UZH, CSG@UZH

[2] Shosh M. Neuman: "Measurement of Technology for National Development", Technological Forecasting and Social Change 55.2 (1999), 119-125.

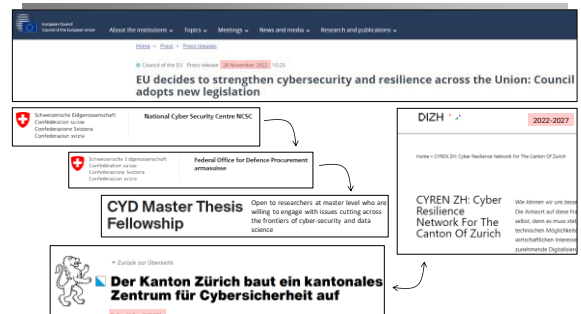
5

[3] Peter Utterton: "The Internet as a Tool for Social Development", United Nations Research Institute for Social Development, <https://www.unrisd.org/hsnr?source=discovery&id=34714>



5

Increased Cybersecurity Awareness



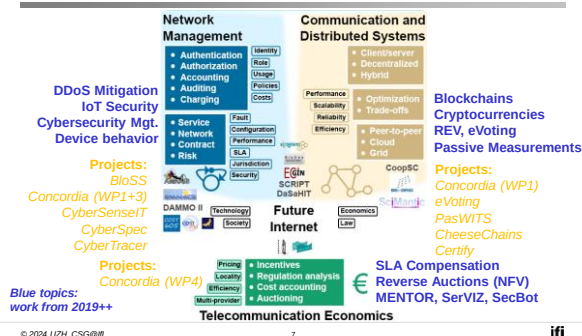
© 2024 UZH, CSG@UZH

6

CYD master thesis ifl

6

CSG's Areas of Work (2004-2022)



© 2024 UZH, CSG@#

7

ifi

7

Blockchain Signaling System (BloSS): Enabling a Cooperative and Multi-domain DDoS Defense

Bruno Rodrigues, Burkhard Stiller
and others

© 2024 UZH, CSG@#

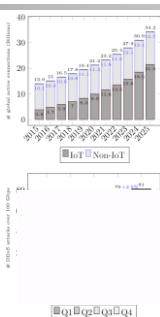
8

ifi

8

Distributed Denial-of-Service (DDoS)

- Rise of DDoS attacks in recent years
 - Increase in the **number of connected devices** and **available bandwidth**
 - **Unsecure** devices exploited by Botnets
- Centralized defense systems are easily overloaded
 - Increasing hardware/software is not ideal
- DDoS defense systems likewise requires **coordination** and **distribution**
 - Reduce the burden of detection/mitigation in a single domain
 - Extend mitigation capabilities



© 2024 UZH, CSG@#

9

inspire neustar sapientia cloud

ifi

9

Challenges for a Cross-organizational Defense

1. **Technical:**
 - Dealing with heterogeneous networking systems
 - Lack of standards for protocols and operations
 - High complexity of deployment and operation
2. **Social:**
 - Confidentiality, public image
 - Management of trust/reputation
3. **Economic:**
 - Mitigating attacks implies in operational expenses
 - Need of incentives
4. **Legal:**
 - Different regulations across different regions

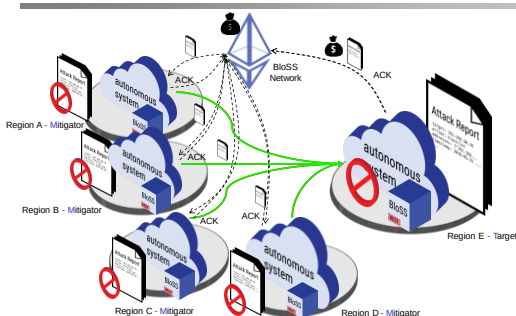
© 2024 UZH, CSG@#

10

ifi

10

BloSS Overview



© 2024 UZH, CSG@#

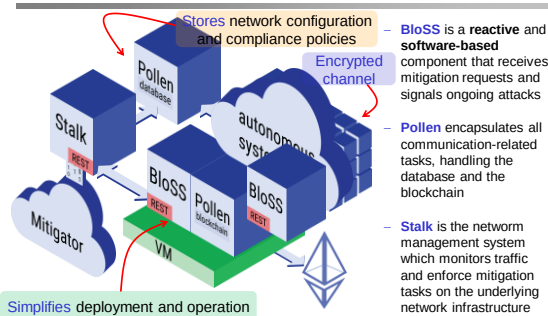
11

Published at AMS 2017 (Full-paper and PhD Track)

ifi

11

BloSS Design



© 2024 UZH, CSG@#

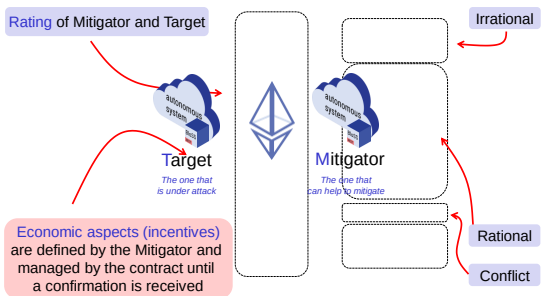
12

Publications: IM 2018 (Full-paper), JNSM 2020 (Journal)

ifi

12

Design of the Cooperative Mitigation Protocol



© 2024 UZH, CSG@ifi

13

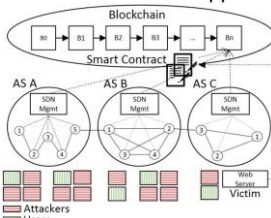
Publications: IEEE CyberSec 2018 (Full paper), IM 2018 (Full paper), ICBC 2020 (Short paper), JNSM 2020 (Journal)

ifi

13

First Prototype (2017-1)

- SDN emulated on Mininet and Blockchain
- Ryu controller + Ethereum Blockchain
- Simple use case to evaluate the approach



© 2024 UZH, CSG@ifi

14

ifi

14

Second Prototype (2017-2)

- Three independent Autonomous Systems (ASes 100, 200, 300)
- Two controllers for each domain and four hosts
- Compute: ASUS Tinker Board (Gb Ethernet)
- Network
 - Zodiac FX OpenFlow switches for inter-domain connection (Fast Ethernet)
 - MikroTik routers for intra-domain (Gigabit Ethernet)



© 2024 UZH, CSG@ifi

15

ifi

15

Prototype Improvements (2018-1)

- PoW → PoA Blockchain
 - No need for mining laptops
- Rockboards 64 for InfluxDB and Graphana
- Modular BloSS with a Rest API



API: Application Programming Interface

© 2024 UZH, CSG@ifi

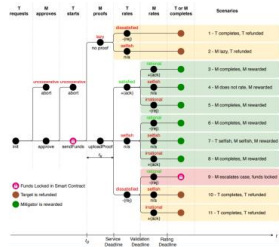
16

ifi

16

Third Prototype (2019-2020)

- Goal: to verify end-to-end latency to complete scenarios
- 11 Scenarios (different ending states of the protocol)
- 4 instances
 - Zürich (UZH)
 - São Paulo (AWS)
 - Ohio (AWS)
 - Tokyo (AWS)



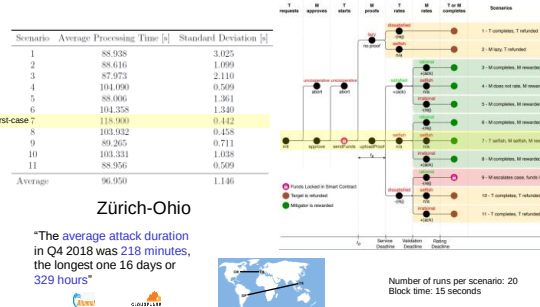
© 2024 UZH, CSG@ifi

17

ifi

17

Evaluations (2019-2020)



© 2024 UZH, CSG@ifi

18

ifi

18

Takeaways

- ❑ Cybersecurity is an **ever-growing concern** present in all major disciplines (software, networked, distributed systems) and topics
 - Always present a tradeoffs where innovation is key
 - Software and hardware must evolve in parallel and in similar levels
 - There is a lot of room for improvement in many fields
- ❑ Research typically focuses in **invention** and **innovation**
 - **innovation** refers to finding optimal tradeoffs and compromises between different parameters
 - Maximizing performance-levels could imply in decreased security-levels
 - **invention** refers to new methods, approaches, and tools to secure specific components
- ❑ Data-oriented approaches are becoming a trend with the excessive amount of [security-related] information being generated
 - Novel ways to **automate security data pipelines**, **data standardization and formatting**, **analysis and representation**, **orchestration of mechanisms**