

Demo

- Getting an SMIME certificate

http://kb.mozillazine.org/Getting_an_SMIME_certificate

- Installing an SMIME certificate

http://kb.mozillazine.org/Installing_an_SMIME_certificate

- Guide to Using S/MIME (Signing and Encrypting Email Messages)

http://www.mozilla.org/projects/security/pki/psm/smime_guide.html

Enigmail örneği :

<https://www.enigmail.net/documentation/screenshots.php>

<https://addons.mozilla.org/en-US/thunderbird/addon/enigmail/>

http://fedoraproject.org/wiki/Using_GPG_with_Thunderbird

Protocols, Web Security, Types of Attacks, Intrusion Detection, Firewall, Some Recommendations

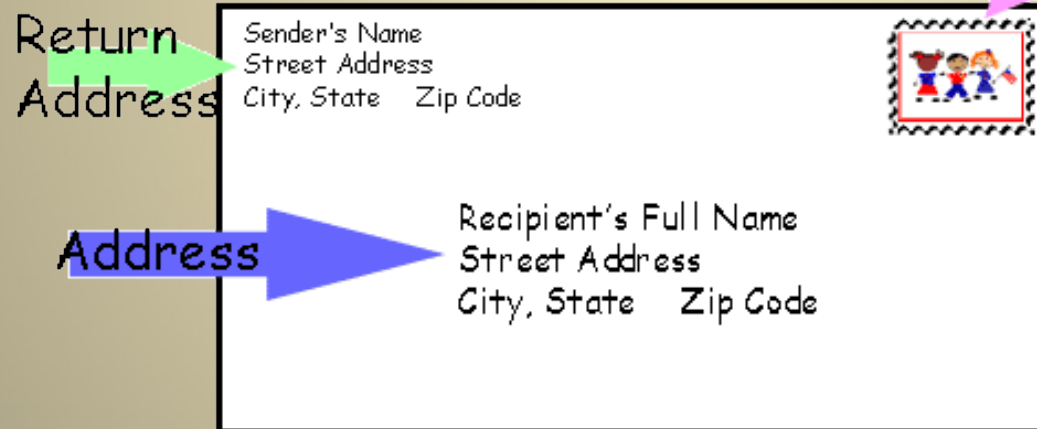
Kayıtlı Elektronik Posta (KEP)

- KEP nedir?
- KEP Protokol Gereksinimleri
- KEP Protokol Çeşitleri
 - Inline
 - Online
 - Offline
 - Örnekler
- Standartlar
- Dünyadaki Uygulamalar

POSTA

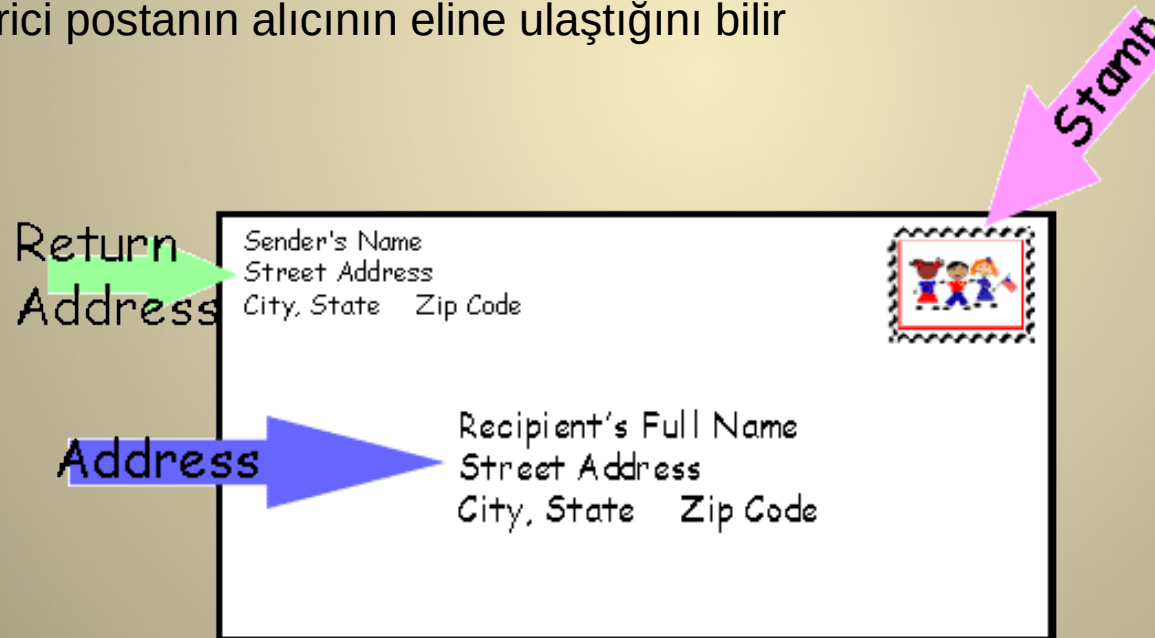
Kim gönderdi?

Posta alıcıya ulaştı mı?



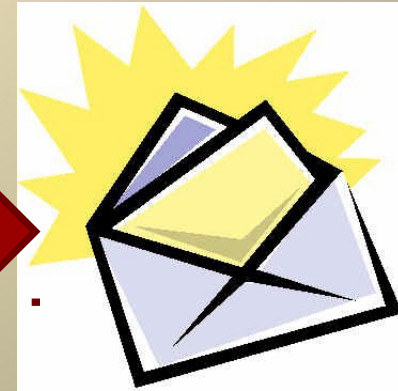
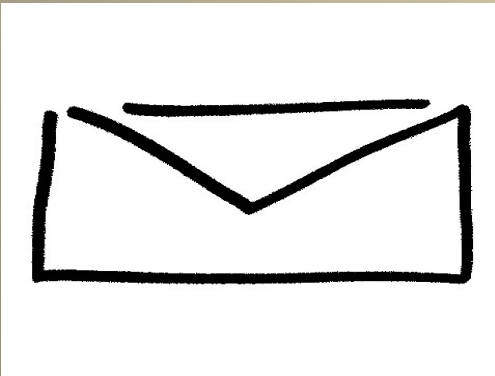
İadeli Taahhütlü posta

- Gönderici postayı iadeli taahhütlü gönderir
- Alıcı postayı aldığı anda imzalar
- Gönderici postanın alıcının eline ulaştığını bilir



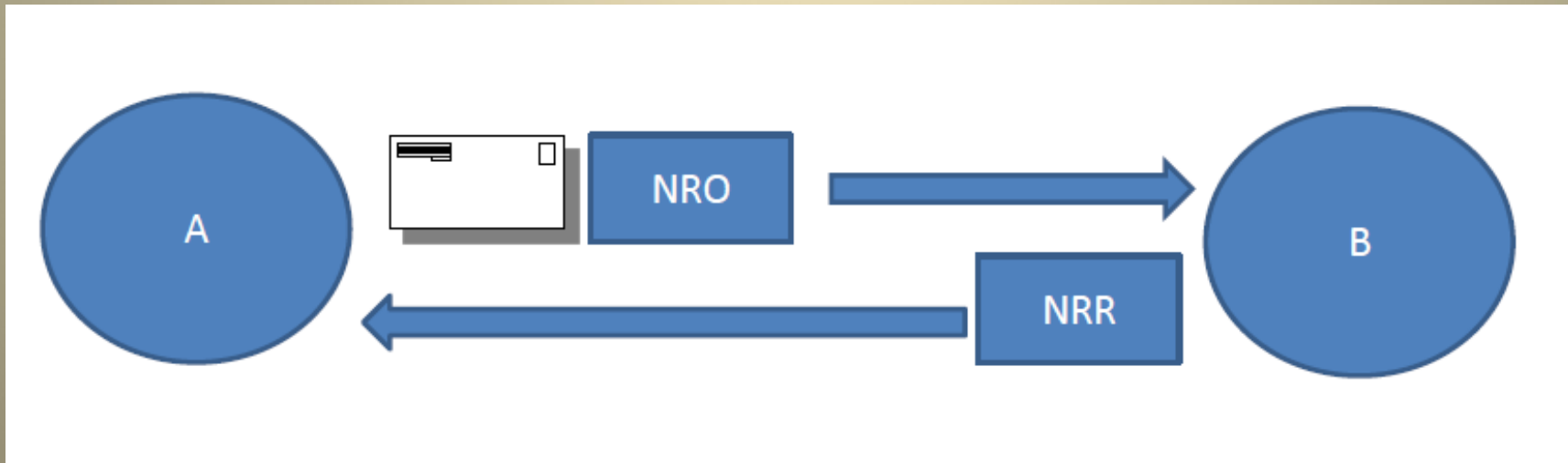
E-posta

- Gönderici gönderdiğini kanıtlayabilir mi?
- Gönderici e-postayı göndermediği halde gönderdiğini iddia edebilir mi?
- Alıcı e-postayı aldığı halde inkar edebilir mi?
- Gizlilik nasıl sağlanır?
- Fatura, banka' maliye gibi işlemlerde yasal olarak kullanılabilecek hale nasıl getirilebilir?



Kayıtlı e-posta (kep)

- Kayıtlı e-posta gönderim kaynağına ve alındı bildirimine dair kanıtları içeren mail alışverişidir
- Gönderen taraf
 - e-postanın kendisi tarafından gönderdiğini ispat edebilir
 - E-postanın alıcı tarafa ulaştığını ispat edebilir
- Alan taraf
 - e-postanın kendine ulaşmadığını ispat edebilir
 - E-postanın nereden gönderildiğini ispat edebilir

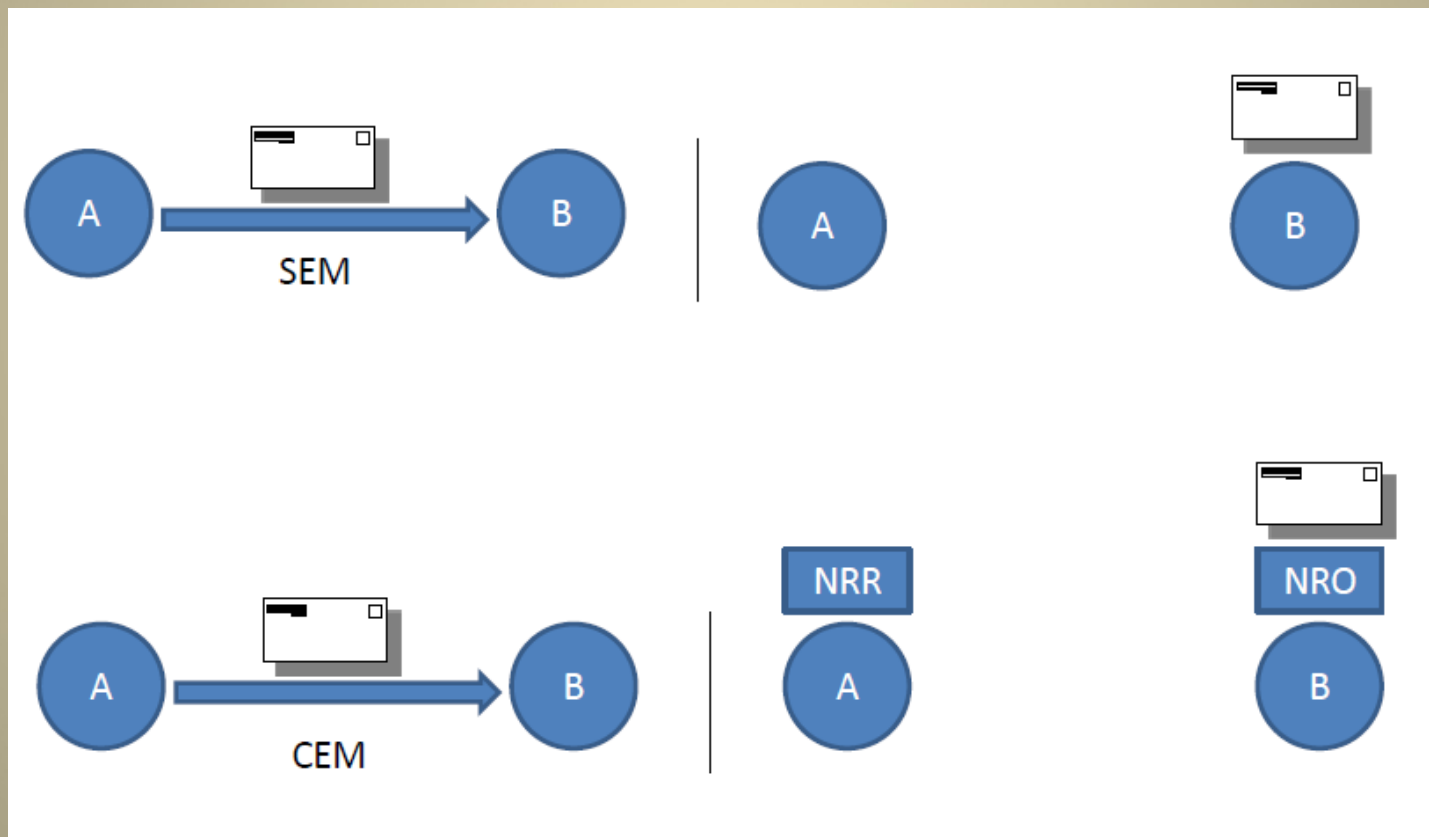


Certified Email Protocols -Kayıtlı Elektronik Posta-

Introduction

- Certified Email Protocols make use of cryptographic algorithms and hence cryptographic protocols (TLS, Diffie-Hellman key exchange, X.509 etc.)
- A cryptographic protocol is an abstract or concrete protocol that performs a security-related function and applies cryptographic methods.
- Cryptographic protocols are widely used for secure application-level data transport.
- Its implementations in Turkey
http://www.tk.gov.tr/bilgi_teknolojileri/kayitli_elektronik_posta/index.php

Standard e-posta ve Kayıtlı e-posta



KEP Protokolü

- E-posta gönderiminin yasal olarakta geçerliliğini sağlayacak, gönderildi ve alındı bildirimleri gibi kanıtları sağlayan temeli kriptografik yapılara dayanan protokol olarak tanımlanabilir.

KEP Protokol Özellikleri

- Adillik (Fairness)
- Gönderildi bildirimi (Sending Receipt)
- Kaynağın inkar edilememesi (Non-repudiation of origin)
- Alındı bildiriminin inkar edilememesi (Non-repudiation of receipt)
- Kimlik doğrulama (Authenticity)
- Bütünlük (Integrity)
- Gizlilik (Confidentiality)
- Zaman aralığı/Yerindelik (Timeliness)
- Zamansal kimlik doğrulama (Temporal Authentication)

Kaynağın inkar edilememesi (Non-repudiation of origin)

- E-posta gönderimini başlatan taraf mesajın kendisi tarafından gönderildiğini inkar edememeli, alıcı taraf mesajın kaynağına dair kanıta sahip olmalı

Alındı bildiriminin inkar edilememesi (Non-repudiation of receipt)

- Alıcı mesajı aldığını inkar edememeli, protokol sonunda gönderici alıcının mesajı aldığını dair bir kanıta sahip olmalı

Adillik (Fairness)

- Protokol adil olmalı: her iki tarafta gerekli bilgileri almalı yada hiçbiri yarar sağlayacak bir bilgiyi elde edememeli

Gönderildi bildirimi (Sending Receipt)

- Gönderici e-posta göndermeyi başlattığına dair kanıt sağlayacak gönderildi bildirimine sahip olmalı

Kimlik doğrulama (Authenticity)

- Protokole katılan taraflar için kimlik doğrulama sağlanmalı

Bütünlük (Integrity)

- Protokole katılan taraflar fark ettirmeden gönderilen mesajların bütünlüğünü bozamamalı

Gizlilik (Confidentiality)

- Sadece gönderici ve alıcı mesaj alışverişleri sonucu orijinal mesaja ulaşabilmeli

Zaman Aralığı (Timeliness)

- Protokol süresi sonlu bir zaman olmalı, belirlenen zaman zarfında protokol sonlanamıyorsa protokol sonlandırılmalı.

Zamansal Kimlik Doğrulama (Temporal Authentication)

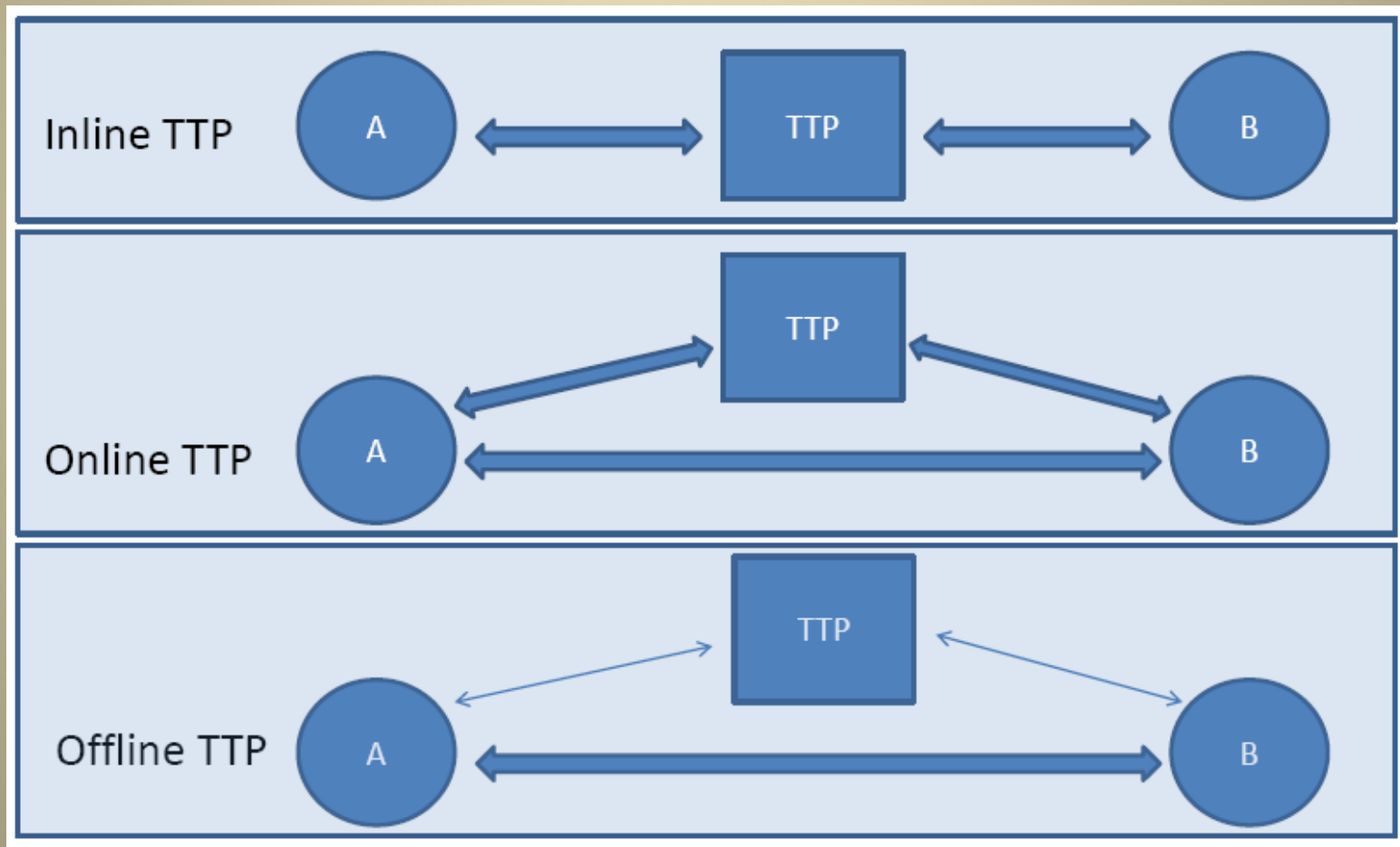
- Mesaj alışverişinin başlama tarihi sertifikalanmalı ve protokoldeki yer alan taraflarca ulaşılabilirmeli

(Zaman Damgası)

KEP Çeşitleri

- Güvenilir Üçüncü Taraf (Trusted Third Noter) içeren yöntemler
 - Inline TTP
 - Çevrim dışı (Offline) TTP
 - Çevrim içi (Online) TTP
- Güvenilir Üçüncü Taraf içermeyen yöntemler

KEP Çeşitleri



Inline TTP

- TTP protokol süresince tüm mesaj transferlerinde teslimat otoritesi olarak görev yapar
- Güçlü adillik kriterlerini sağlar (tüm mesajlar başka adrese yönlendirilmeden önce TTP üzerinden geçer; TTP gerekli tüm bilgiyi toplar)
- TTP protokol akışında tam kontrole sahiptir (zamansal kimlik doğrulama)
- TTP'nin yoğun bir şekilde dahil olması sebebiyle iletişim ve hesaplama darboğazına neden olur

Online TTP

- TTP'nin dahiliyetini azaltır
- Online TTP keş protokolünün her oturumunda yarar alır ancak her adımına dahil olmaz
- Görevi kriptografik anahtarlar ve/veya alındı bildirimleri gibi protokol işaretlemelemelerinin idame ettirmektir

Offline TTP

- Inline ve Online TTP performansını arttırmaya yönelik tasarlanmıştır
- TTP olağandışı durumlarda devreye girer (herhangi bir tarafın aldatma eylemi , iletişimin kopması)
- Aldatma
 - Alıcı mesajı aldığı halde inkar edebilir
 - Gönderici geçerli bir alındı bildirimi almasına rağmen alıcıya orjinal mesajı vermeyebilir
- TTP aldatma durumlarında aldatan tarafın bir avantaj elde edemeyeceğini sağlamakla yükümlüdür
- Çoğu zaman protokolün normal olarak çalışacağı varsayılmıştır: *Optimistic Protocol*

Protokollerin karşılaştırılması

TABLE I: Comparison of the properties of CEM protocols

Property	[7]	[14]	[13]	[31]	[29]	[16]	[5]	[8]
TTP Involvement	Inline	Inline	Inline	Inline	Online	Online	Offline	Offline
Fairness	✓	✓	✓	✓	✓	✓	✓	✓
Confidentiality		✓	✓	✓	✓			
Timeliness	✓		✓		✓		✓	
Transparent							✓	✓
Verifiable	✓	✓	✓	✓	✓	✓		
Stateful	✓	✓	✓	✓	✓	✓		
Stateless							✓	✓

Bahreman and Tygar Protokolünün Genişletilmiş Sürümü (Inline TTP)

- Stelvio Cimato, Clemente Galdi, Raffaella Giordano, Barbara Masucci ve Gildo Tomasco tarafından öne sürülmüştür.
- Bahremen ve Tygar'ın öne sürdüğü protokole ek olarak zaman damgası fonksiyoneliyesi eklenmiştir.
- KEP gereksinimlerinin hepsini karşıladığı öne sürülmüştür.

Protokol Tanımları

- $\text{SignA}(m)$: m mesajının A tarafından imzalanmış hali
- $h(.)$: Özet fonksiyon
- $\text{PKB}(m)$: m mesajının B'nin açık anahtarı ile şifrelenmiş hali
- $\text{Ek}(m)$: m mesajının k gizli anahtarı ile şifrelenmiş hali

In-line TTP CEP by Cimato et al. (2005)

Derived from the first in-line TTP certified email protocol (Bahreman and Tygar, 1994)

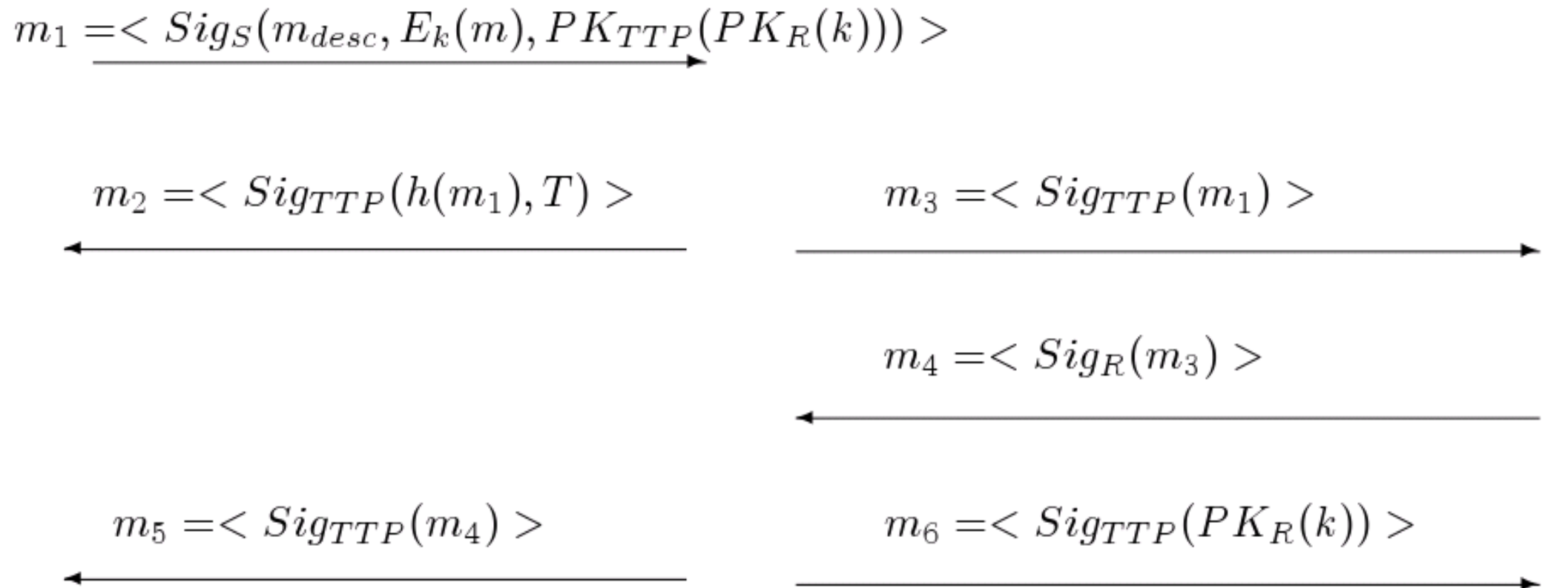
The goal of the protocol is to allow a sender *S* to send an e-mail message to a receiver *R*

Sender

Trusted Third Party

Receiver

$m_1 = \langle \text{Sig}_S(m_{desc}, E_k(m), PK_{TTP}(PK_R(k))) \rangle$



$m_2 = \langle \text{Sig}_{TTP}(h(m_1), T) \rangle$

$m_3 = \langle \text{Sig}_{TTP}(m_1) \rangle$

$m_4 = \langle \text{Sig}_R(m_3) \rangle$

$m_5 = \langle \text{Sig}_{TTP}(m_4) \rangle$

$m_6 = \langle \text{Sig}_{TTP}(PK_R(k)) \rangle$

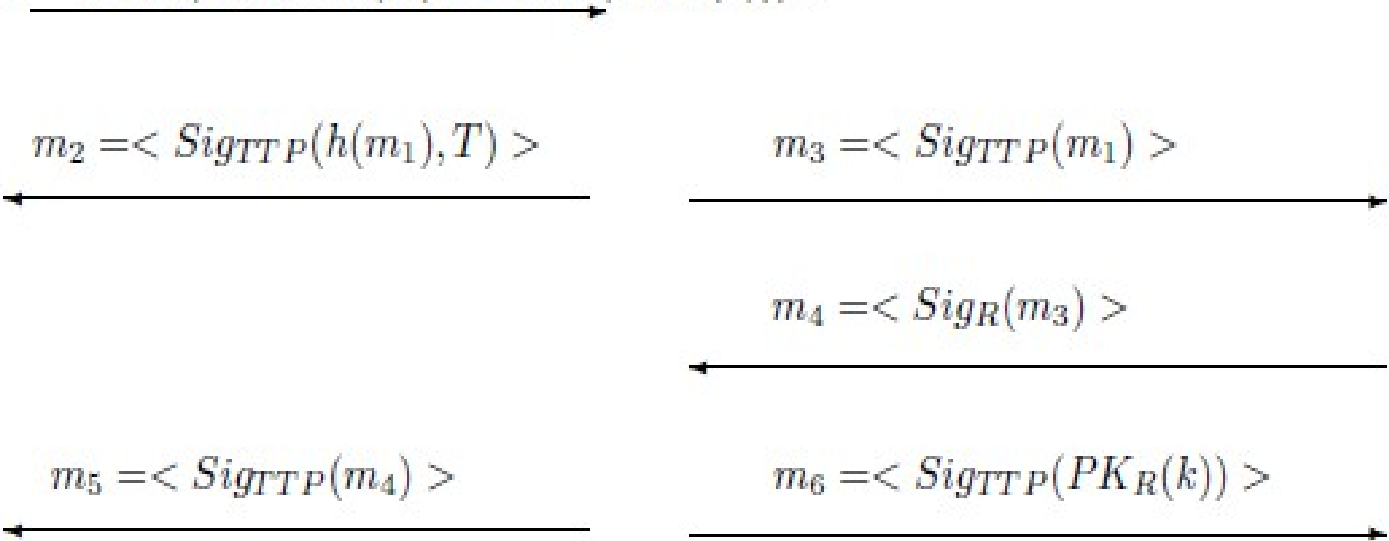
Properties: Fairness, Sending Receipt, NRO, NRR, Authenticity, Integrity, Confidentiality, Timeliness, Temporal Authentication

Protokol İşleyişi

Gönderici

TTP

Alıcı

$$m_1 = \langle \text{Sig}_S(m_{desc}, E_k(m), PK_{TTP}(PK_R(k))) \rangle$$


$$m_2 = \langle \text{Sig}_{TTP}(h(m_1), T) \rangle$$

$$m_3 = \langle \text{Sig}_{TTP}(m_1) \rangle$$

$$m_4 = \langle \text{Sig}_R(m_3) \rangle$$

$$m_5 = \langle \text{Sig}_{TTP}(m_4) \rangle$$

$$m_6 = \langle \text{Sig}_{TTP}(PK_R(k)) \rangle$$

GG Protokolü (Offline TTP)

- Galdi and Giordano tarafından öne sürülmüştür
- Geliştirilmiş optimistik bir kayıtlı e-posta protokolüdür
- Temel protokol zamansallık ve mesaj doğrulanabilirliğini desteklemektedir.
- Gönderici ve alıcı arasında üç mesaj değişimi yapılmaktadır.
- Protokoldeki ana fikir mesajın elektronik bir zarfla kilitlenmesidir
- Genişletilmiş versiyonunda zamansal kimlik doğrulama için ek bir mesaj yer almaktadır.

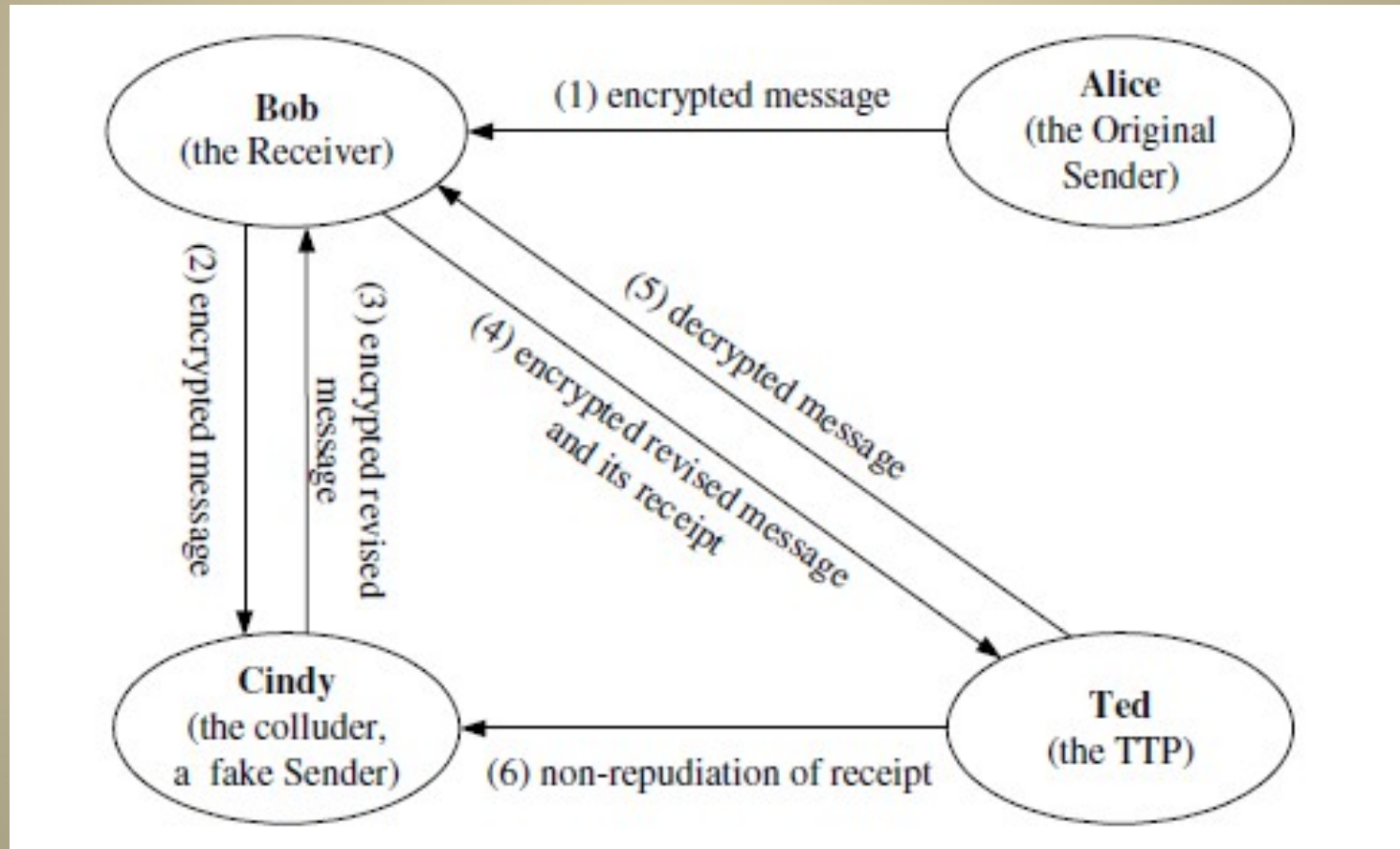
GG Protokol Tanımları

- Alice : Gönderen taraf
- Bob: Alan taraf
- Ted : Offline TTP
- Sam: TSS
- Msubj : m mesajının konu bölümü
- PKX(m): X'in açık anahtarı ile m mesajının şifrelenmesi
- Signx(m): X'in mesaj m üzerindeki imzası
- PKX(m,r) : X'in açık anahtarı ve rassal r ile şifrelenen mesaj m
- X->Y : m m mesajının X'ten Y'ye gönderilmesi
- X||Y : x ve y dizgilerinin ardarda bağlanması
- H(.) : Özet fonksiyon

Protokol İşleyişi

- (e1). Alice \rightarrow Sam: m_1 ,
where $m_1 = \langle env, Sig_A(env) \rangle$,
 $env = \langle ID_A, ID_B, PK_B(\langle m_{subj}, h(\langle m, r \rangle) \rangle),$
 $\overline{PK}_T(PK_B(msg), r) \rangle, msg = \langle m_{subj}, m \rangle$
- (e2). Sam \rightarrow Bob and/or Alice: m_2 ,
where $m_2 = \langle \langle m_1, t(m_1) \rangle, Sig_S(\langle m_1, t(m_1) \rangle) \rangle$
- (e3). Bob \rightarrow Alice: m_3 ,
where $m_3 = \langle m_2, Sig_B(m_2) \rangle$
- (e4). Alice \rightarrow Bob: m_4 ,
where $m_4 = \langle \langle PK_B(msg), r \rangle, Sig_A(\langle PK_B(msg), r \rangle) \rangle$

GG Protokol Atakları



OS Protokolü (Online TTP)

- Oppliger and Stadlin tarafından alıcı ve göndericinin TTP ile olan etkileşimini azaltmaya yönelik tasarlanmıştır.
- Mesaj anahtarını kayıtlı e-postaya dual-kriptografik imza ile bağlamıştır.
- Temel protokolde
 - Gönderici biri alıcıya diğeri TTP'ye olmak üzere iki mesaj gönderir
 - Alıcı TTP'den mesaj için anahtar ister
 - TTP alıcıya anahtarı göndericiye de alındı bildirimini gönderir

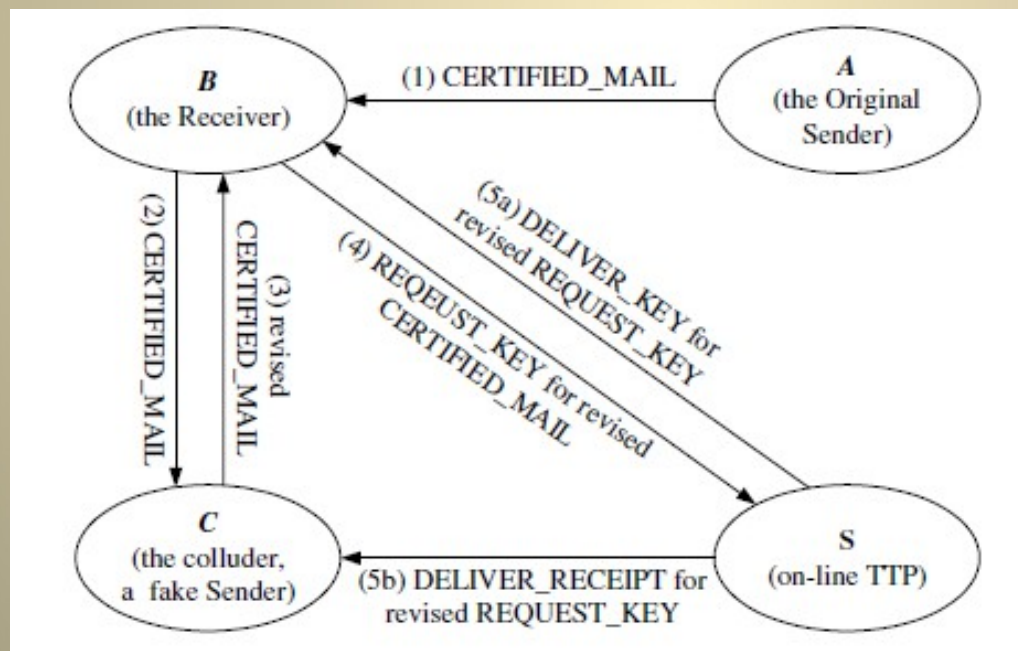
Protokol Tanımları

- A: Gönderici taraf
- B: Alıcı Taraf
- S: TTP
- M: Mesaj, M' M mesajinin kriptografik olarak şifrelenmiş değeri, M'' M' mesajinin rassal bir anahtarla değeri
- K: Gizli anahtar (KA A'nın gizli anahtarı)
- (k, k^{-1}) açık anahtar ikilisi
- $\{M\}_K, \{M\}_k$: gizli anahtarla/açık anahtarla şifrelenmiş M mesajının değeri
- $h()$ özet fonksiyon
- $\{h(h(M1), h(M2))\}_{k^{-1}}$ dual imza

Protokol İşleyişi

- 1: $A \rightarrow_{\text{SMTP}} B$: CERTIFIED_MAIL $((\langle A, B, S, M'', \{\{K\}k_B\}k_S, \{h(h(M''), h(\{\{K\}k_B\}k_S))\}k_A^{-1})k_A^{-1})$
- 2: $B \Rightarrow_{\text{HTTPS}} S$: REQUEST_KEY $((\langle A, B, h(M''), \{\{K\}k_B\}k_S, \{h(h(M''), h(\{\{K\}k_B\}k_S))\}k_A^{-1})k_B^{-1})$
- 3a: $S \Rightarrow_{\text{HTTPS}} B$: DELIVER_KEY $((\langle A, B, h(M''), \{K\}k_B\}k_S^{-1})$
- 3b: $S \rightarrow_{\text{SMTP}} A$: DELIVER_RECEIPT $((\langle A, B, S, h(M''), h(\{K\}k_B), \{h(h(M''), h(\{K\}k_B))\}k_S^{-1})k_S^{-1})$

OS Protokol Atakları



Certified Email Protocols

Some measurements and Design principles to avoid attacks

- **Guideline 1 (Shao et al, 2005):** It should be clearly set out the assumptions or requirements on communication channels in a certified email scheme (and maybe all security protocols), so that others rather than the original designers can properly understand, evaluate and implement the scheme.
- **Guideline 2 (Shao et al, 2005):** In the design of certified email schemes, the exploited cryptographic algorithms should be specified clearly on the aspects of security levels and operational features. In some scenarios, not all functionally equivalent cryptographic primitives can be used to implement the proposed scheme.

Certified Email Protocols

Some measurements and Design principles to avoid attacks

- **Guideline 3 (Shao et al, 2005):** Since non-repudiation evidences are obviously linked to the involved parties, the identities of the originator, the recipient, and maybe the TTP, should be embedded in those evidences explicitly or implicitly.
- **Guideline 4 (Shao et al, 2005):** As the pivotal component of a certified email scheme (and all fair exchange protocols), the dispute resolution policy should present detailed descriptions on the definition, explanation and verification of all non-repudiation evidences.

Certified Email Protocols

Some measurements and Design principles to avoid attacks

- **Design Principle 1 (Gürgens et al. 2005):** For a label (eg. $H(A, B, TTP, H(C), H(K))$, where H is a cryptographic hash function) that is supposed to identify a certain exchange transaction, security properties such as
 - Verifiability,
 - Uniqueness and
 - Secrecy can be taken into account.
- **Design Principle 2 (Gürgens et al. 2005):**
 - **Authenticity of messages:** All message parts should be included in the respective signature.
 - **Verifiability of messages:** Every recipient of a message should be able to verify this message.
 - **Context of messages:** It should be possible for the recipient of a message to identify the transaction to which its parts belong.

Certified Email Protocols

Some measurements and Design principles to avoid attacks

- **Design Principle 3 (Gürgens et al. 2005)**

Requirements for the TTP behavior:

- **Meaningful TTP decisions**
- **Reply to every request**

- **Design Principle 4 (Kremer et al. 2002)**

- The security of a non-repudiation protocol also depends on some ad-hoc problems.
- One of the most important issues is good management of the non-repudiation evidences, the used digital signatures and the corresponding keys.
- It can happen that a secret signature generation key is compromised. It is then necessary to revoke the certificate of the corresponding public verification key.
- It is necessary to be able to identify whether a signature was generated before or after the revocation. **Solution: Use of Timestamps.**

Certified Email Protocols

Some measurements and Design principles to avoid attacks

- **Design Principle 4 (Louridas 2000)**
 - Match protocols and requirements.
 - Be careful with the termination problem.
 - Be careful with implementation details.
 - Formal verifications highlight assumptions.
 - Be careful with practical improvements. It is possible that , in the process, problems will be introduced.
 - Treat adjudication as an integral part of the protocol. Adjudication should be well defined and deterministic, and guarantee a single yes or no answer for each set of input. Protocol analysis should include adjudication analysis.
 - Assumptions should always be explicit.

Certified Email Protocols and for Other Security Protocols

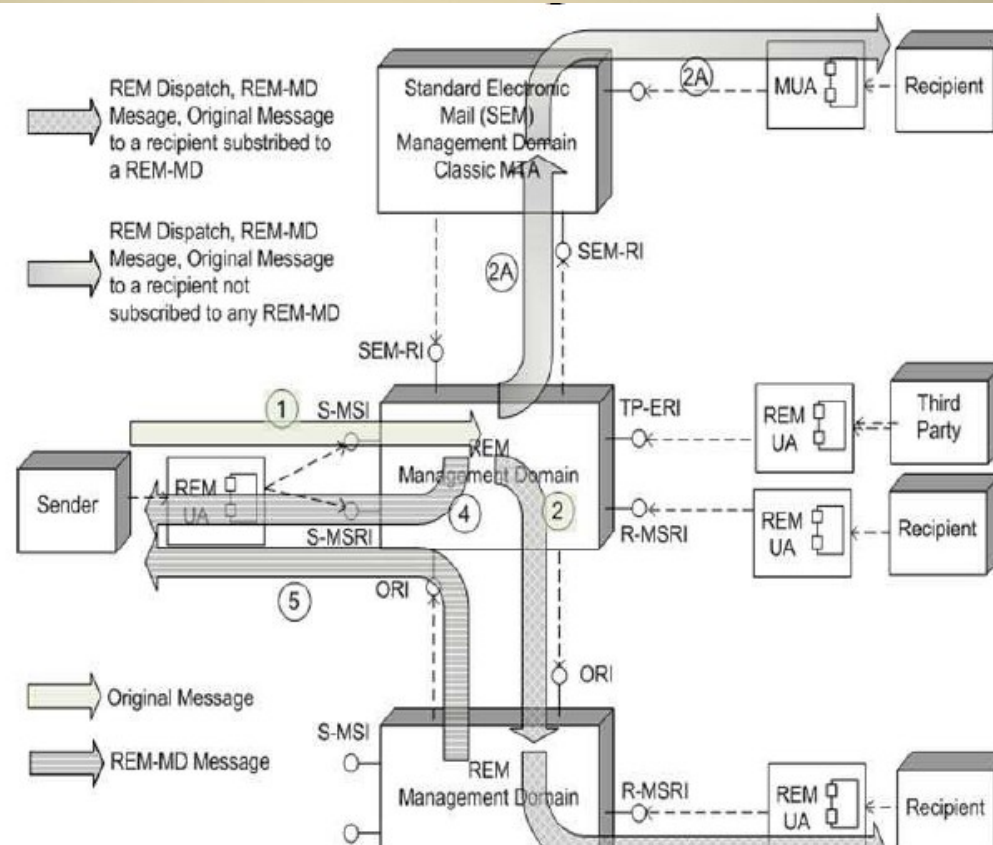
Issues on their analysis

- **Standard Model**
- **Provable Security Models (Random oracle model etc.)**
- **Formal Methods (pi-calculus etc.)**
- **Verification of security properties by tools and languages (automatic verifiers)**
- **Probabilistic model checking**
- **Finite-state checking**

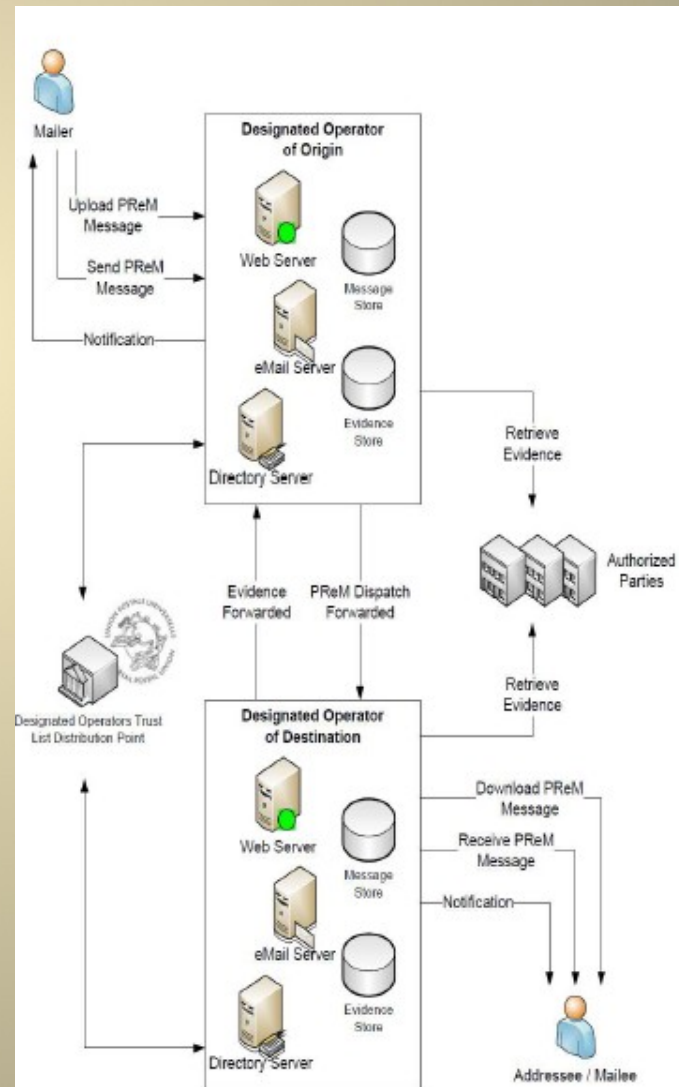
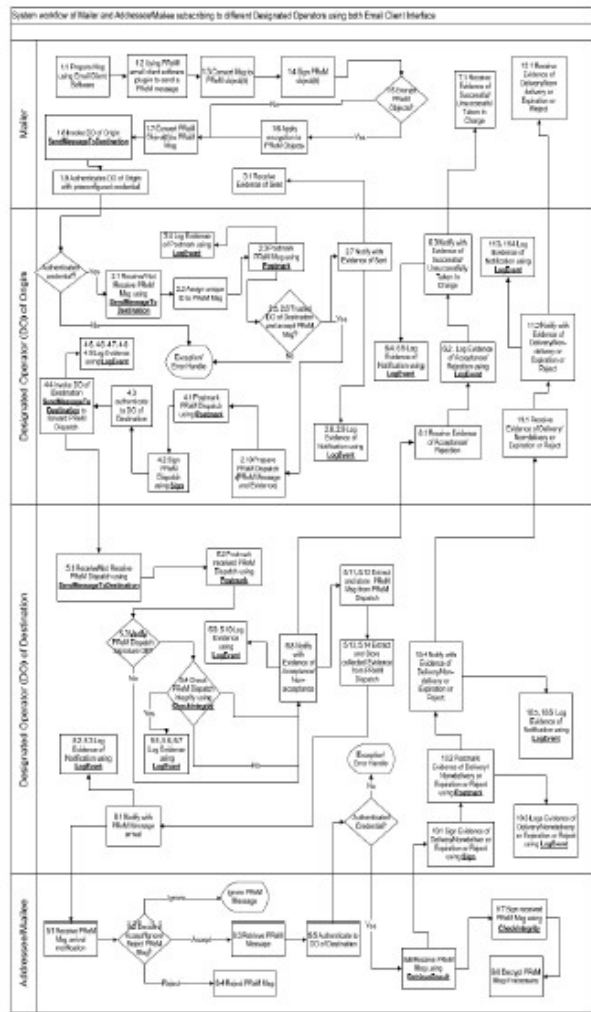
KEP Standartları

- ETSI TS 102 640-1,2,3,4,5 V 1.1.1(2010-01)
Electronic Signatures and Infrastructures
(ESI); Registered Electronic Mail (REM) ;
Architecture Format and Policies
- IETF (draft) Certified Electronic Mail
- UNIVERSAL POSTAL UNION, Postal
Registered e-Mail (PReM) Functional
Specification v0.52

ETSI



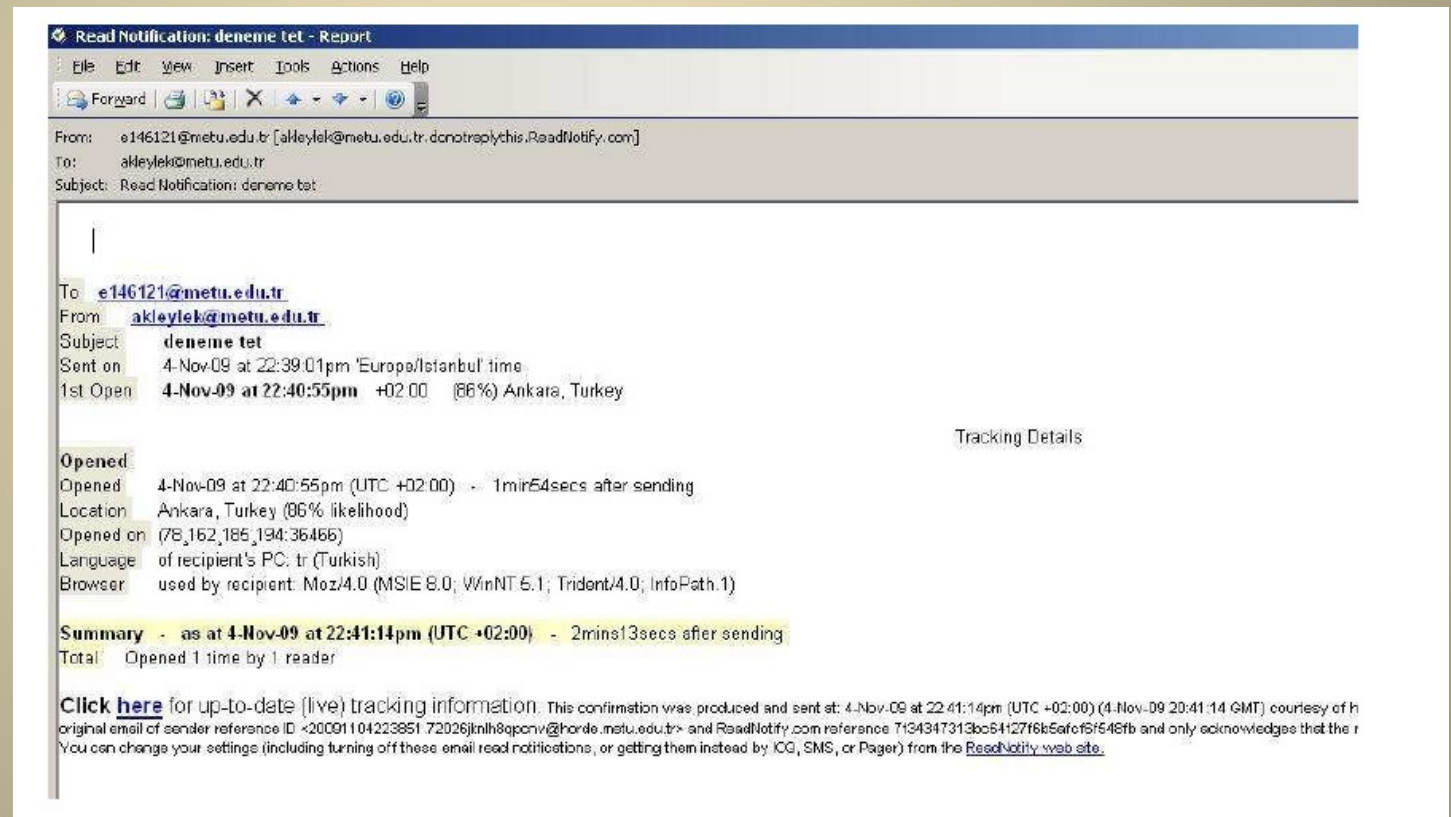
UPU



Dünya'daki Uygulamalar

- Inline
 - ReadNotify
 - Posta Elettronica Certificata
 - CertifiedMail
 - ZixMail
- Online
 - GoodMail
 - Datamotion
 - Rewpost
- Hybrid (Online-Offline)
 - Tricert

ReadNotify



Read Notification: deneme tet - Report

File Edit View Insert Tools Actions Help

Forward

From: e146121@metu.edu.tr [akylek@metu.edu.tr.dcnoreplychis.ReadNotify.com]
To: akylek@metu.edu.tr
Subject: Read Notification: deneme tet

To: e146121@metu.edu.tr
From: akylek@metu.edu.tr
Subject: deneme tet
Sent on: 4-Nov-09 at 22:39:01pm 'Europe/Istanbul' time
1st Open: 4-Nov-09 at 22:40:55pm +02:00 (86%) Ankara, Turkey

Tracking Details

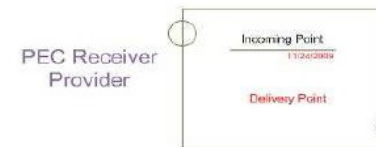
Opened

Opened: 4-Nov-09 at 22:40:55pm (UTC +02:00) - 1min54secs after sending
Location: Ankara, Turkey (86% likelihood)
Opened on: (7B,162,185,194:36466)
Language: of recipient's PC: tr (Turkish)
Browser: used by recipient: Moz/4.0 (MSIE 8.0; WinNT 5.1; Trident/4.0; InfoPath.1)

Summary - as at 4-Nov-09 at 22:41:14pm (UTC +02:00) - 2mins13secs after sending
Total: Opened 1 time by 1 reader

Click [here](#) for up-to-date (live) tracking information. This confirmation was produced and sent at: 4-Nov-09 at 22:41:14pm (UTC +02:00) (4-Nov-09 20:41:14 GMT) courtesy of the original email of sender reference ID <20091104223851.72026jnlh8qpcv@honda.metu.edu.tr> and ReadNotify.com reference T134347313bc64127f6b5afcf5f648fb and only acknowledges that the r
You can change your settings (including turning off these email read notifications, or getting them instead by ICQ, SMS, or Pager) from the [ReadNotify web site](#).

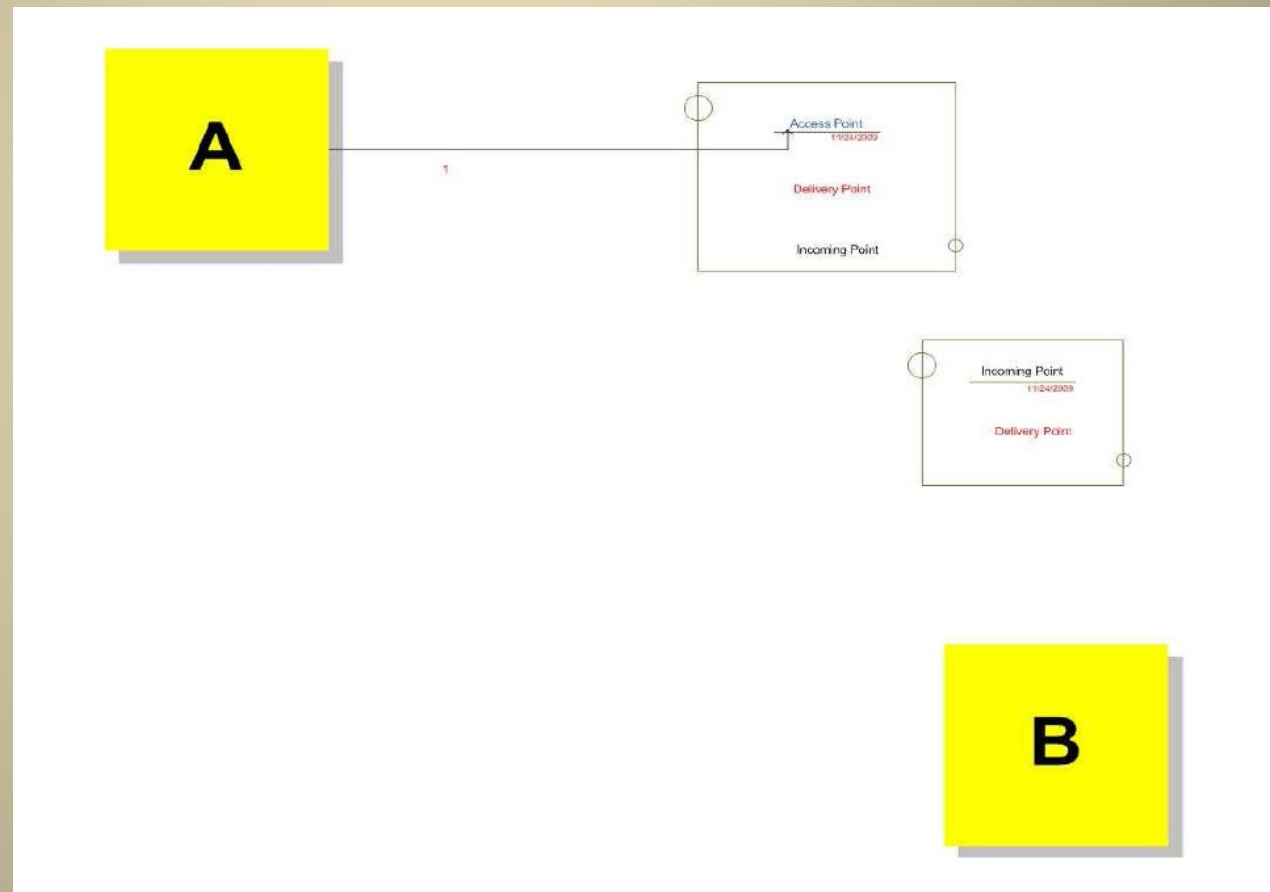
Posta Elettronica Certificata



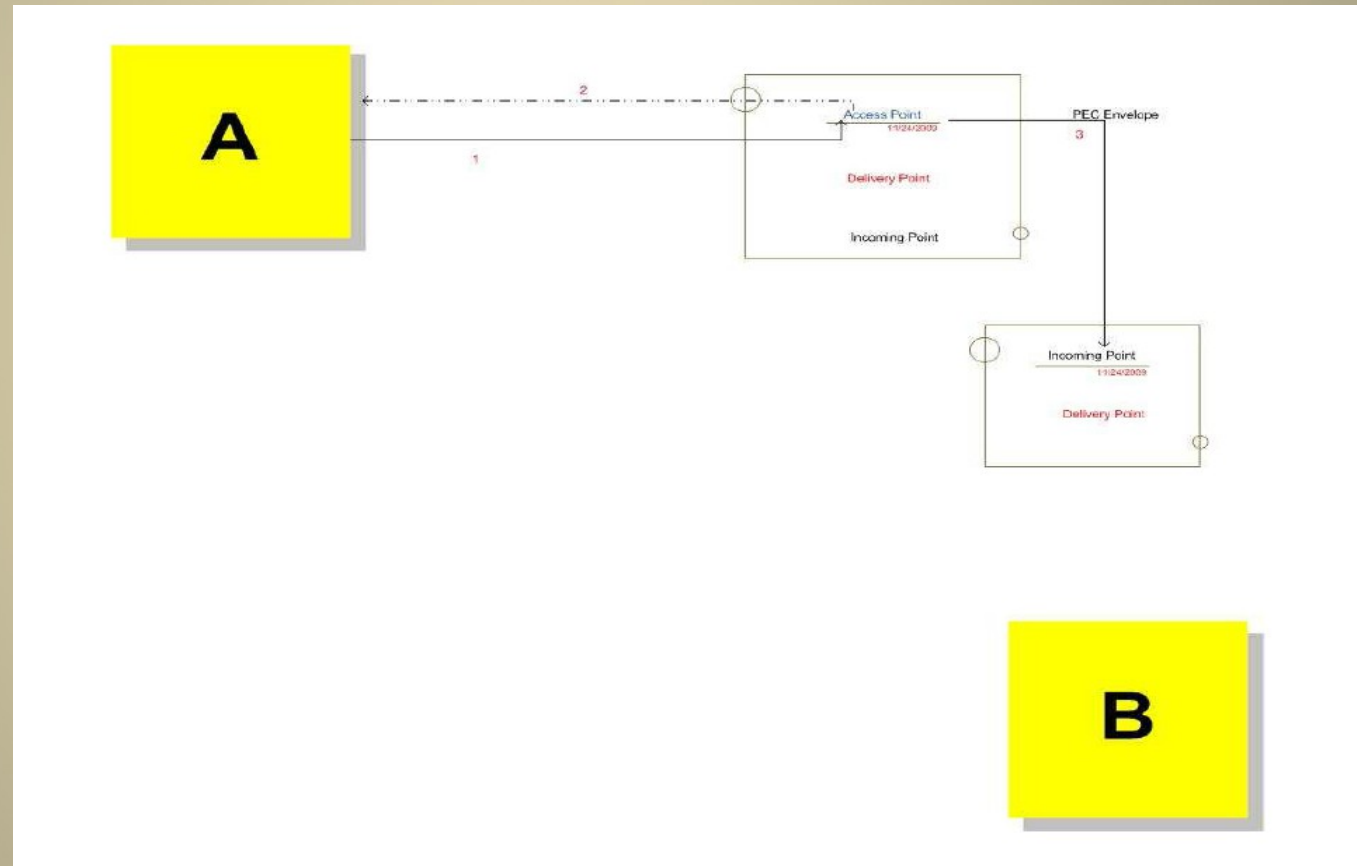
Posta Elettronica Certificata



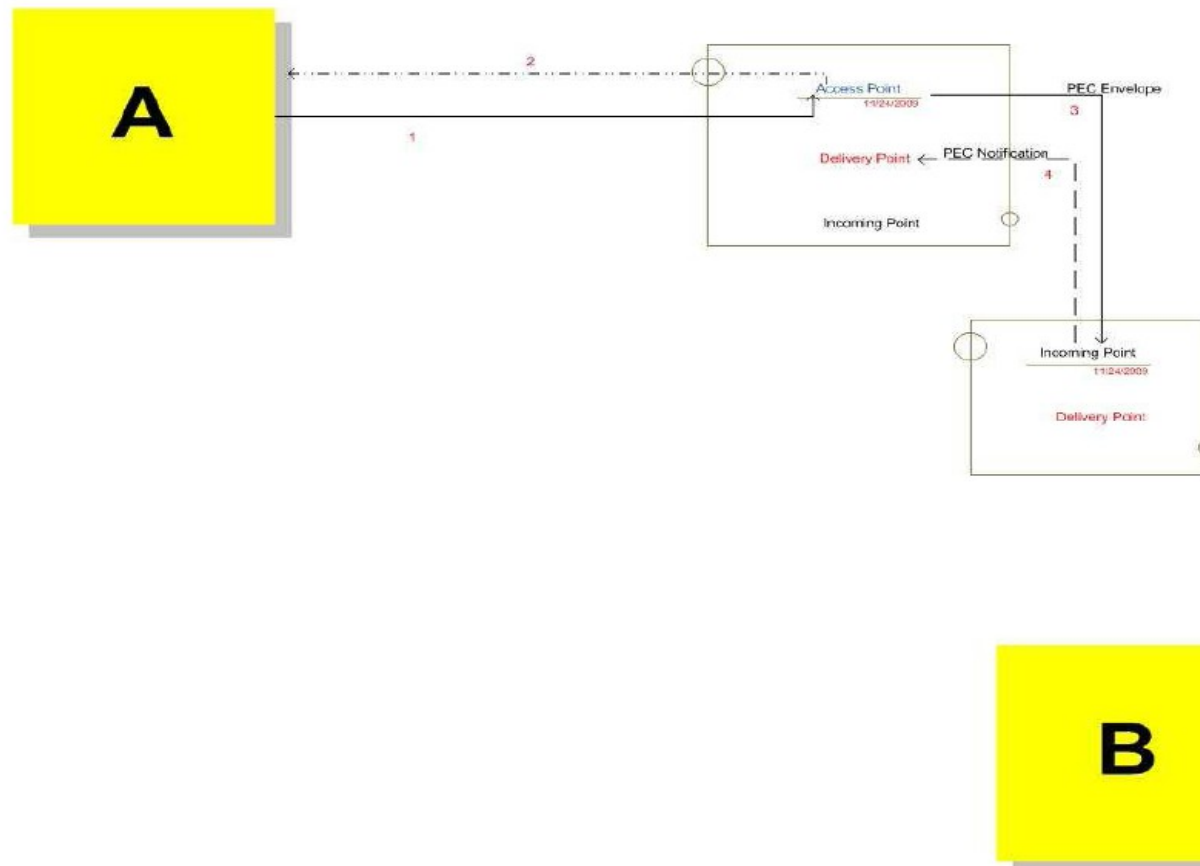
Posta Elettronica Certificata



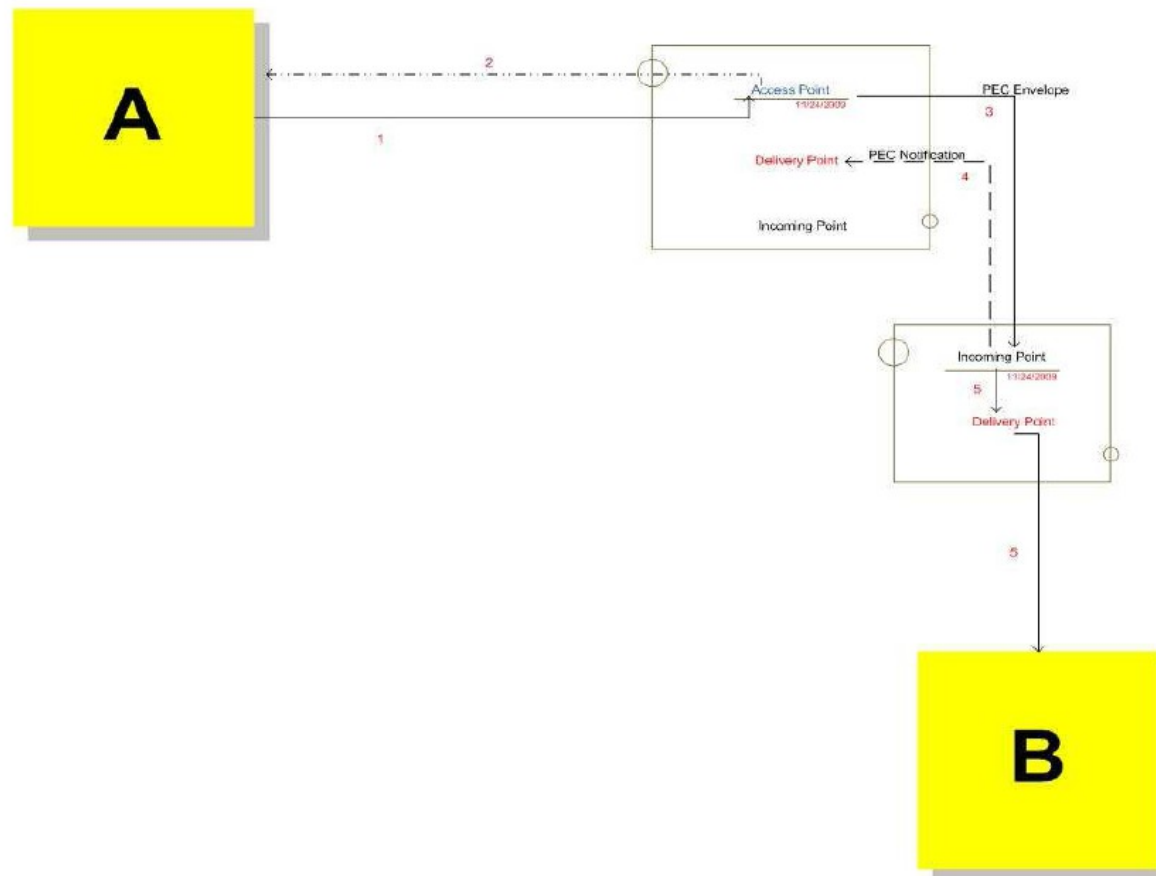
Posta Elettronica Certificata



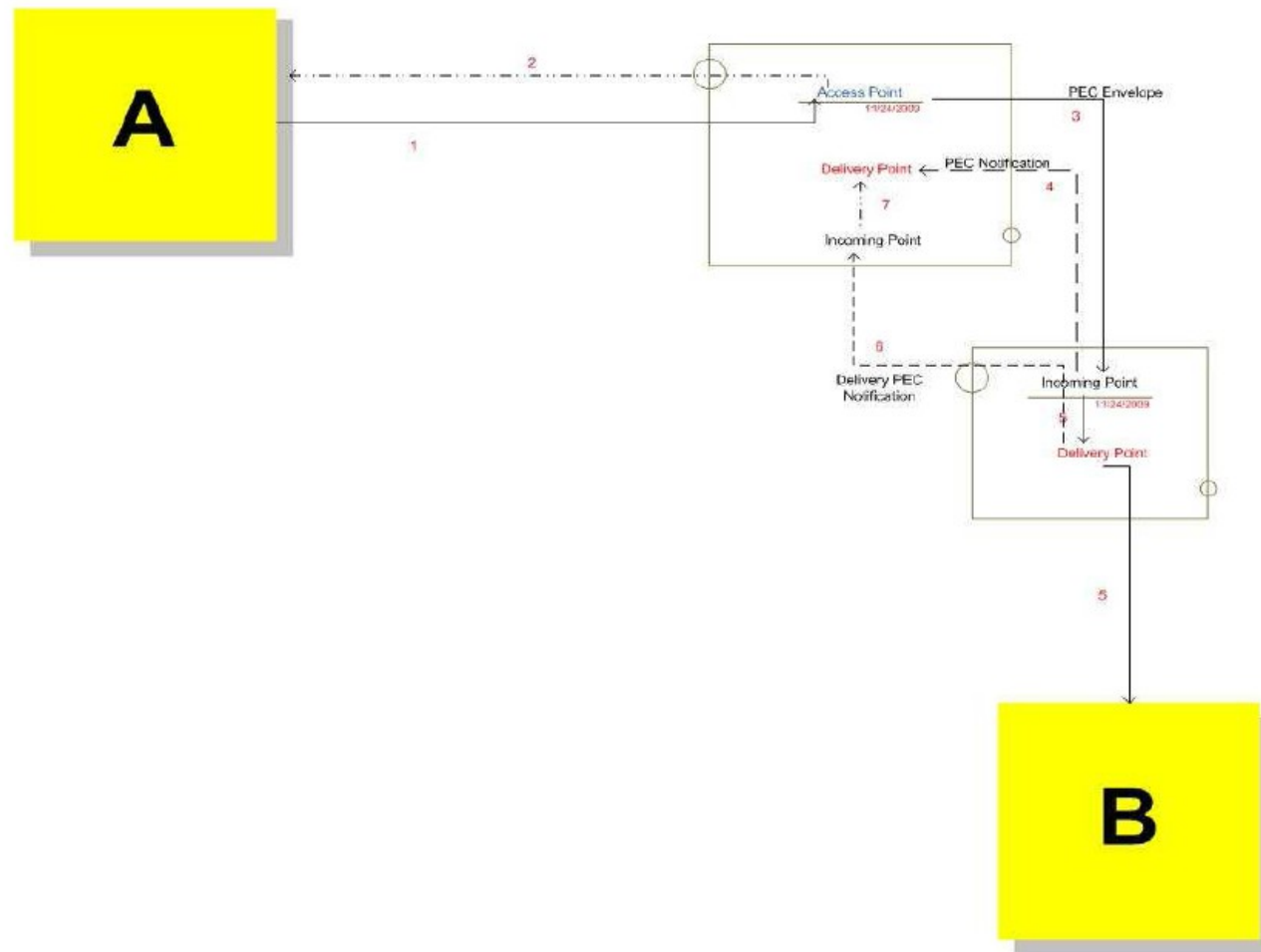
Posta Elettronica Certificata



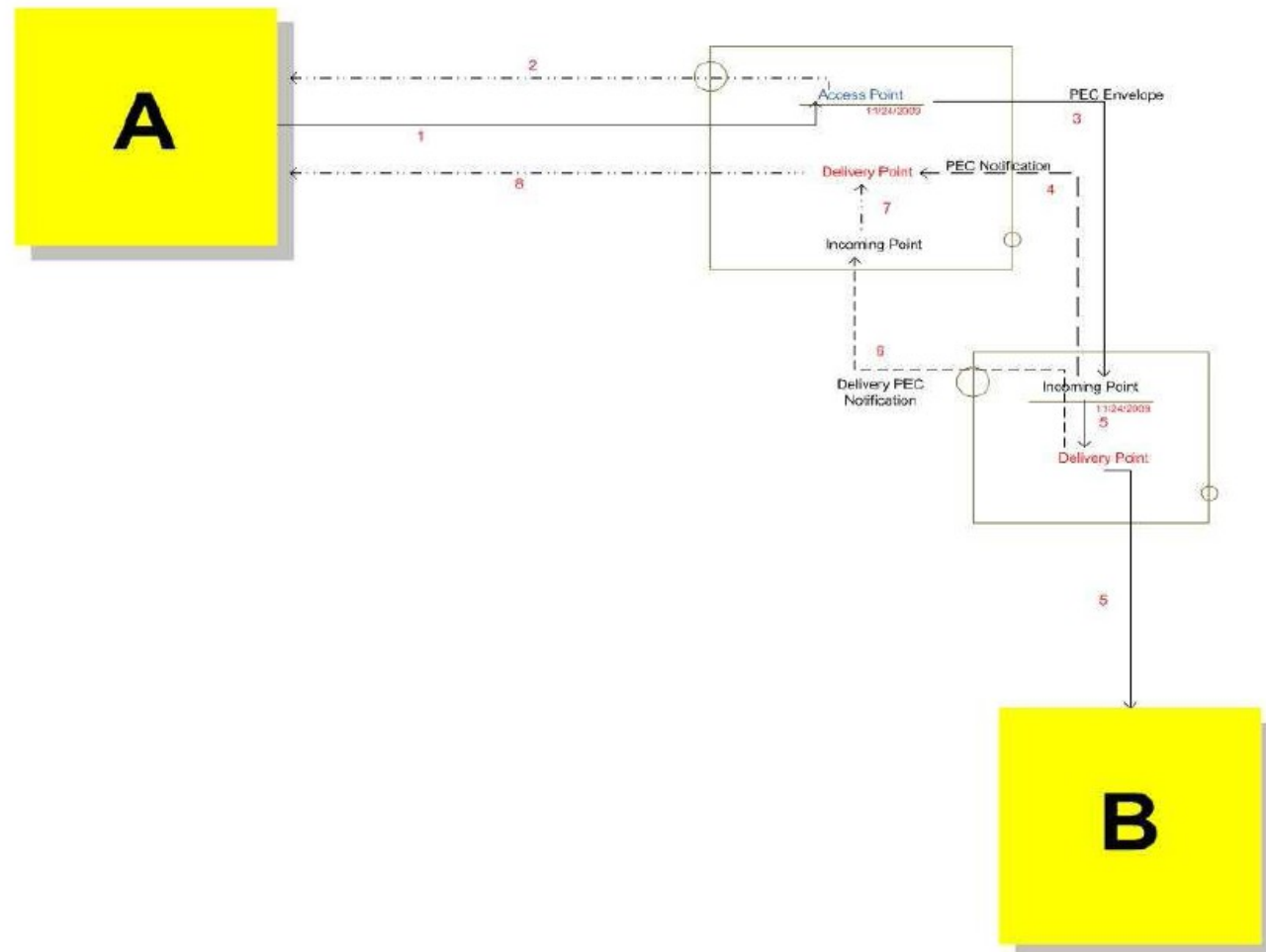
Posta Elettronica Certificata



Posta Elettronica Certificata



Posta Elettronica Certificata



Posta Elettronica Certificata

- E –İmza
 - Hardware Security Module
- Kimlik Doğrulama
 - Şifre
 - Elektronik Kimlik Kartı
- KEP Servis Sağlayıcılarının Haberleşmesi
 - Güvenilir Kanallar ve Protokoller
 - TLS
- S/MIMI Sertifikası
 - X509

Posta Elettronica Certificata

```
VERSION: 3
SERIAL: 11226 (0x2bda)
INNER SIGNATURE:
  ALG. ID: id-sha1-with-rsa-encryption
  PARAMETER: 0
ISSUER:
  Country Name: IT
    Organization Name: Certifier 1
    Organizational Unit Name: Certification Service Provider
    Common Name: Certifier S.p.A.
VALIDITY:
  Not Before: Oct 5, 04 09:04:23 GMT
  Not After: Oct 5, 05 09:04:23 GMT
SUBJECT:
  Country Name: IT
    Organization Name: AcmePEC S.p.A.
    Common Name: Certified Mail

PUBLIC KEY: (key size is 1024 bits)
ALGORITHM:
ALG. ID: id-rsa-encryption
PARAMETER: 0
MODULUS: 0x00afbe4 5563198a aa9bac3f 1b29b5be
          7f691945 89d01569 ca0d555b 5c33d7e9
          ...
          d15ff128 6792def5 b3f884e6 54b326db
          cf
EXPONENT: 0x010001
EXTENSIONS:
  Subject Alt Name:
  RFC Name: posta-certificata@acmepec.it
  Key Usage*: Digital Signature
  Authority Key Identifier: 0x12345678 aaaaaaaa bbbbbbbb
cccccccc

dddddddd
| Subject Key Identifier: 0x3afae080 6453527a 3e5709d8 49a941a8

a3a70ae1
|SIGNATURE:
  ALG. ID: id-sha1-with-rsa-encryption
  PARAMETER: 0
  VALUE: 0x874b4d25 70a46180 c9770a85 fe7923ce
          b22d2955 2f3af207 142b2aba 643aaa61
          ...
          d8fd10b4 c9e00ebc c089f7a3 549a1907
          ff885220 ce796328 b0f8ecac 86ffb1cc
```

Posta Elettronica Certificata

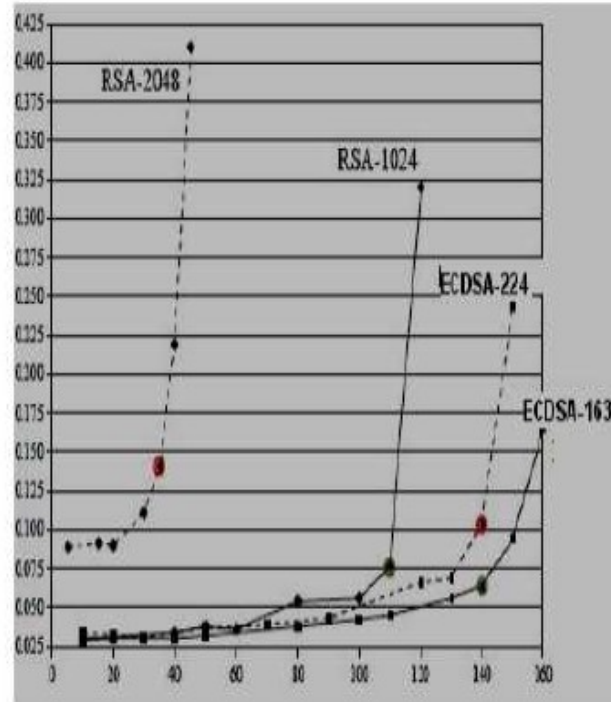
opec3 - signing + receipts

Processes	Time
1	Test completion in 46" - Average: 2,6 sec/email
2	Test completion in 29" - Average: 1,6 sec/email
4	Test completion in 29" - Average: 1,6 sec/email
8	Test completion in 29" - Average: 1,6 sec/email
16	Test completion in 29" - Average: 1,6 sec/email

opec3 - verification + receipts

Processes	Time
1	Test completion in 3'20" - Average: 11,11 sec/email
2	Test completion in 2'11" seconds - Average: 7,28 sec/email
4	Test completion in 2' seconds - Average: 6,67 sec/email
8	Test completion in 1'55" - Average: 6,39 sec/email
16	Test completion in 1'55" - Average: 6,39 sec/email

ECDSA ve RSA Karşılaştırılması

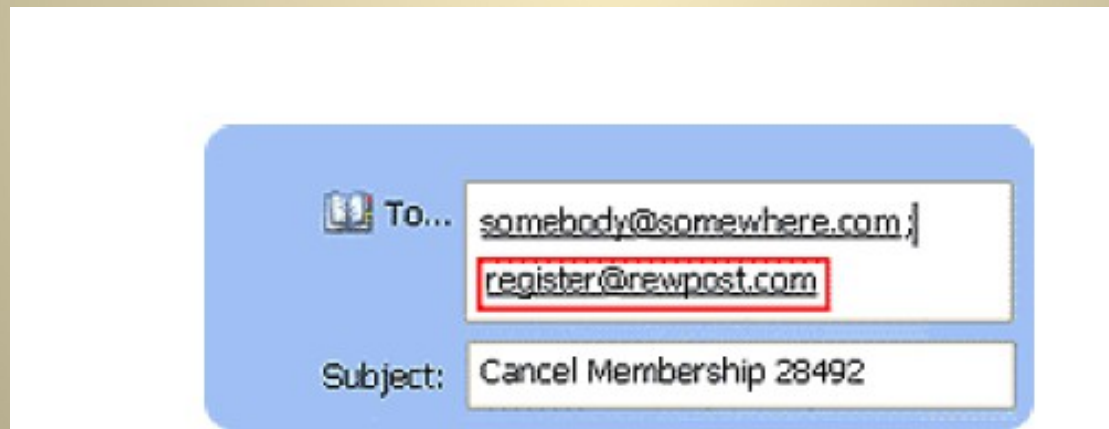


Şekil 1 : Sunucunun cevap verme süresi(ms) *
Saniyede bağlanan kullanıcı sayısı [Sun04]

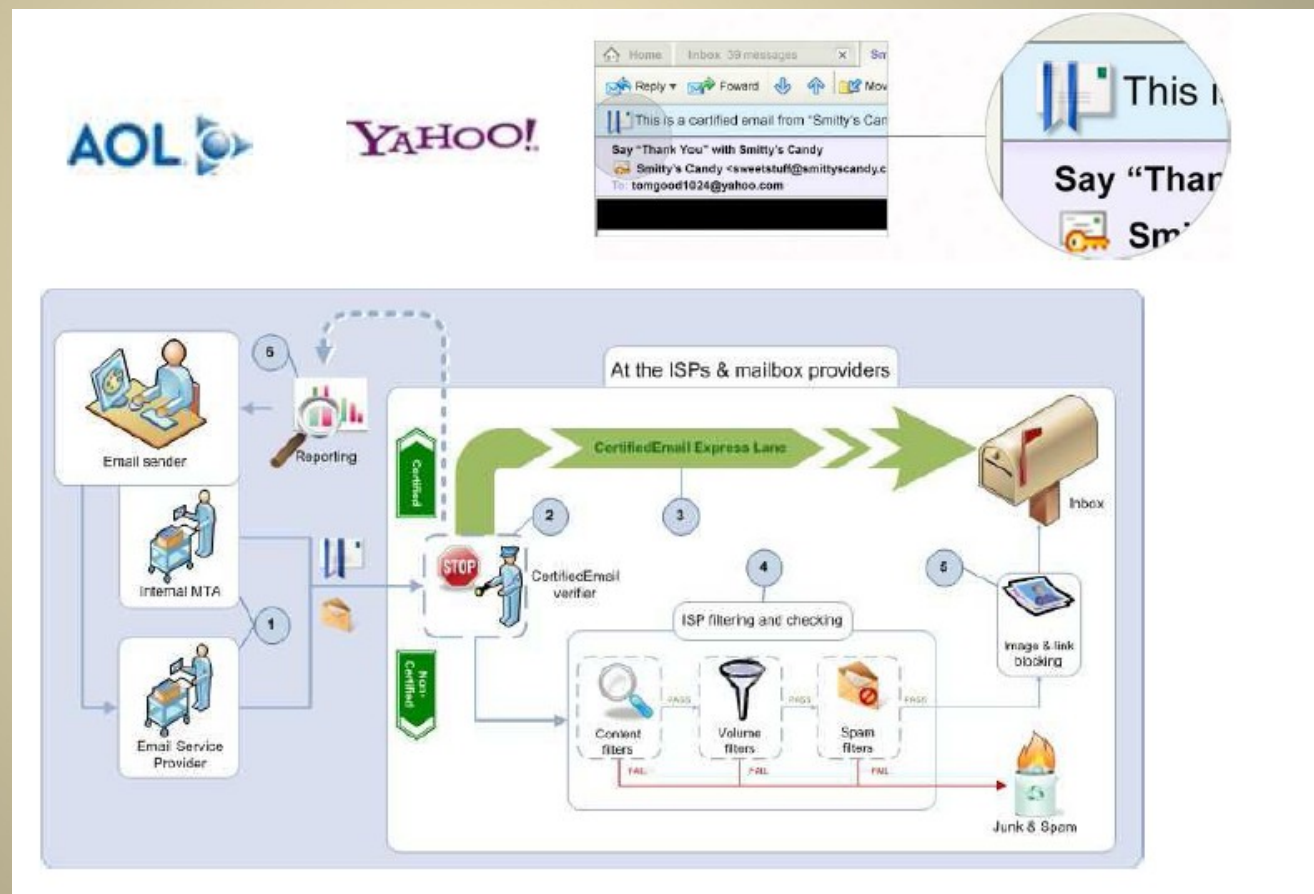
	ECDSA163 RSA1024		ECDSA224 RSA2048	
Süre(ms)	3.69	8.75	5.12	56.18
oran	1/2.4		1/11	

Datamotion ve Rewpost

- FIPS 140-2 standardına uyumlu



GoodMail



Karşılaştırma

	Online (Goodmail)	Inline (ReadNotify)
Ücret	0.25 cent	1 cent

Protocols Security

- Scyther (<http://www.cs.ox.ac.uk/people/cas.cremers/scyther/>) is a tool for the automatic verification of security protocols.

For a performance comparison between Scyther and a number of tools developed by others, please read the abstract of each paper:

- ASICS: Authenticated Key Exchange Security Incorporating Certification Systems
- Evaluation of ISO/IEC 9798 Protocols
- Provably Repairing the ISO/IEC 9798 Standard for Entity Authentication
- Automated Analysis of Diffie-Hellman Protocols and Advanced Security Properties
- Key Exchange in IPsec revisited: Formal Analysis of IKEv1 and IKEv2

<http://www.cs.ox.ac.uk/people/cas.cremers/publications/index.html>

Protocols Security

- ProVerif: Cryptographic protocol verifier in the formal model
<http://prosecco.gforge.inria.fr/personal/bblanche/proverif/>
- ProVerif is an automatic cryptographic protocol verifier, in the formal model
 - It can handle many different cryptographic primitives, including shared- and public-key cryptography (encryption and signatures), hash functions, and Diffie-Hellman key agreements, specified both as rewrite rules or as equations.
- AVISPA stands for Automated Validation of Internet Security Protocols and Applications (<http://www.avispa-project.org/>)

Protocols Security

-References recommended -

- Internet Security Protocols:

<http://www.youtube.com/watch?v=CZzd3i7Bs2o>

- ADVANCING AUTOMATED SECURITY PROTOCOL VERIFICATION (Thesis)

<http://e-collection.library.ethz.ch/eserv/eth:7011/eth-7011-02.pdf>

- Security Protocol Verification: Symbolic and Computational Models

<http://cs.ioc.ee/etaps12/invited/blanchet-slides.pdf>

- SECURE KEY MANAGEMENT PROTOCOL IN WIMAX

https://www.idc-online.com/technical_references/pdfs/data_communications/SECURE%20KEY.pdf

- Enhanced Mobile SET Protocol with Formal Verification

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6394714

SELinux http://en.wikipedia.org/wiki/Security-Enhanced_Linux

- Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides the mechanism for supporting access control security policies, including United States Department of Defense-style mandatory access controls (MAC).
- SELinux is a set of kernel modifications and user-space tools that can be added to various Linux distributions. Its architecture strives to separate enforcement of security decisions from the security policy itself and streamlines the volume of software charged with security policy enforcement. The key concepts underlying SELinux can be traced to several earlier projects by the United States National Security Agency.
- It has been integrated into the Linux kernel mainline since version 2.6, on 8 August 2003.

SeLinux Features http://en.wikipedia.org/wiki/Security-Enhanced_Linux

- Clean separation of policy from enforcement
- Well-defined policy interfaces
- Support for applications querying the policy and enforcing access control (for example, crond running jobs in the correct context)
- Independent of specific policies and policy languages
- Independent of specific security label formats and contents
- Individual labels and controls for kernel objects and services
- Support for policy changes
- Separate measures for protecting system integrity (domain-type) and data confidentiality (multilevel security)
- Flexible policy
- Controls over process initialization and inheritance and program execution
- Controls over file systems, directories, files, and open file descriptors
- Controls over sockets, messages, and network interfaces
- Controls over use of "capabilities"
- Cached information on access-decisions via the AVC (Access Vector Cache)

Web Security Considerations

- A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex
- Casual and untrained (in security matters) users are common clients for Web-based services
- Such users are not necessarily aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures
- The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets
- The following characteristics of Web usage suggest the need for tailored security tools:
 - Web servers are relatively easy to configure and manage
 - Web content is increasingly easy to develop
 - The underlying software is extraordinarily complex
 - May hide many potential security flaws

Table 6.1 A Comparison of Threats on the Web

	Threats	Consequences	Countermeasures
Integrity	<ul style="list-style-type: none"> •Modification of user data •Trojan horse browser •Modification of memory •Modification of message traffic in transit 	<ul style="list-style-type: none"> •Loss of information •Compromise of machine •Vulnerability to all other threats 	Cryptographic checksums
Confidentiality	<ul style="list-style-type: none"> •Eavesdropping on the net •Theft of info from server •Theft of data from client •Info about network configuration •Info about which client talks to server 	<ul style="list-style-type: none"> •Loss of information •Loss of privacy 	Encryption, Web proxies
Denial of Service	<ul style="list-style-type: none"> •Killing of user threads •Flooding machine with bogus requests •Filling up disk or memory •Isolating machine by DNS attacks 	<ul style="list-style-type: none"> •Disruptive •Annoying •Prevent user from getting work done 	Difficult to prevent
Authentication	<ul style="list-style-type: none"> •Impersonation of legitimate users •Data forgery 	<ul style="list-style-type: none"> •Misrepresentation of user •Belief that false information is valid 	Cryptographic techniques

HTTP	FTP	SMTP
TCP		
IP/IPSec		

(a) Network Level

HTTP	FTP	SMTP
SSL or TLS		
TCP		
IP		

(b) Transport Level

	S/MIME	
Kerberos	SMTP	HTTP
UDP	TCP	
IP		

(c) Application Level

Figure 6.1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

SSL , TLS http://en.wikipedia.org/wiki/Transport_Layer_Security

- **Transport Layer Security (TLS)** and its predecessor, **Secure Sockets Layer (SSL)**, are cryptographic protocols which are designed to provide communication security over the Internet.
- They use X.509 certificates and hence asymmetric cryptography to assure the counter party with whom they are communicating, and to exchange a symmetric key.
- This session key is then used to encrypt data flowing between the parties. This allows for data/message confidentiality, and message authentication codes for message integrity and as a by-product, message authentication.
- Several versions of the protocols are in widespread use in applications such as web browsing, electronic mail, Internet faxing, instant messaging, and voice-over IP (VoIP).
- An important property in this context is forward secrecy, so the short term session key cannot be derived from the long term asymmetric secret key.

SSL, TLS

- Applications use TLS or SSL to establish secure connections between two communicating parties. The primary goal of both protocols is to provide privacy and data integrity. Other goals are as follows:
 - Enabling interoperability between applications
 - Providing an extensible framework that can readily incorporate new public key and bulk encryption methods
 - Ensuring relative computational efficiency
- Both TLS and SSL comprise two layers: a Record Protocol and a Handshake Protocol.
- Although the two protocols are similar, the differences are sufficiently significant that SSL 3.0 and the various versions of TLS do not interoperate.
- **Recommended Video: Using SSL/TLS**
<http://www.youtube.com/watch?v=RPvqpjLqcbQ>

- One of the most widely used security services
- A general purpose service implemented as a set of protocols that rely on TCP
 - Could be provided as part of the underlying protocol suite and therefore be transparent to applications
 - Can be embedded in specific packages

SSL and TLS

- SSL was developed by Netscape and is now used by most browsers including Microsoft
- TLS working group was formed within IETF
- First version of TLS can be viewed as an SSLv3.1 and became Internet Standard
- Uses TCP to provide reliable end to end service

SSL Architecture

- SSL is not a single protocol, but two layers of protocols:
- SSL Record Protocol provides basic security services to higher layer protocols, especially to HTTP
- 3 higher layer protocols are part of SSL:
 - Handshake protocol
 - Change Cipher Spec protocol
 - Alert protocol

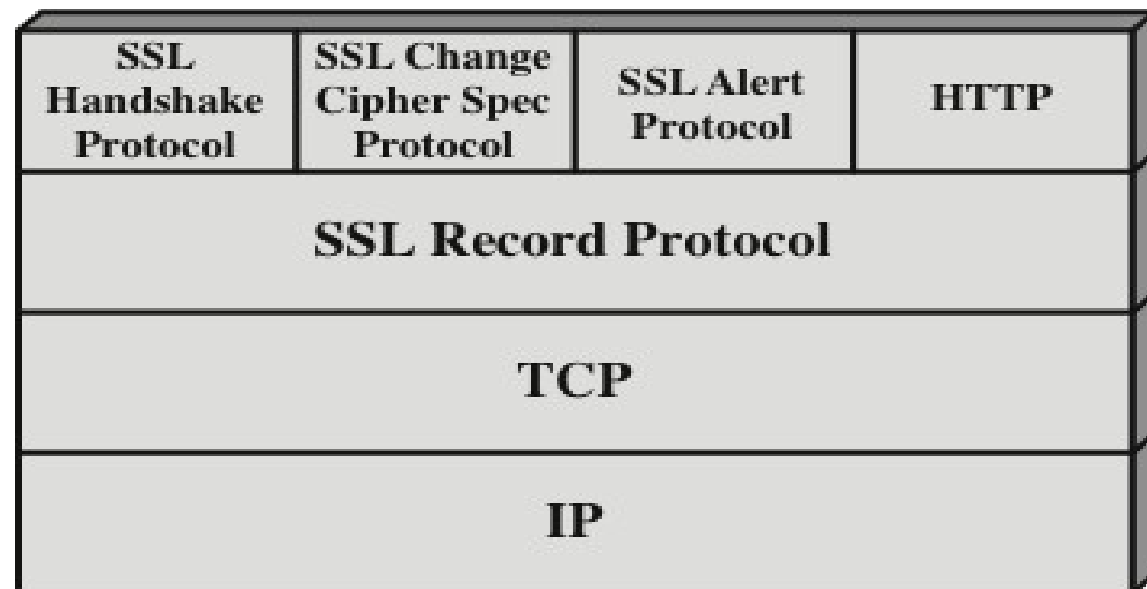


Figure 6.2 SSL Protocol Stack

SSL Architecture

- **SSL connection**
 - connections are transient, peer-to-peer
 - every connection associated with one SSL session
- **SSL session**
 - an association between client and server
 - created by the Handshake Protocol
 - define a set of cryptographic parameters
 - may be shared by multiple SSL connections

SSL Session State

- A Session state is defined by:
 - Session identifier-id for session state
 - Peer Certificate - X509.v3 certificate
 - Compression Method - used before encryption
 - Cipher Space- data encryption algorithm
 - Master Secret- shared between client and server
 - Is resumable- can initiate new connection

SSL Connection State

- A connection state is defined by:
- Server and Client random
- Server write MAC secret
- Client write Mac secret
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

A connection state is defined by the following parameters.

- **Server and client random:** Byte sequences that are chosen by the server and client for each connection.
- **Server write MAC secret:** The secret key used in MAC operations on data sent by the server.
- **Client write MAC secret:** The secret key used in MAC operations on data sent by the client.

A connection state is defined by the following parameters:

- **Server write key:** The secret encryption key for data encrypted by the server and decrypted by the client.
- **Client write key:** The symmetric encryption key for data encrypted by the client and decrypted by the server.
- **Initialization vectors:** When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter, the final ciphertext block from each record is preserved for use as the IV with the following record.
- **Sequence numbers:** Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero.

SSL Record Protocol

SSL Record Protocol defines two services for SSL connections:

- **Confidentiality:** The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads. The message is compressed before being concatenated with the MAC and encrypted, with a range of ciphers being supported as shown.
- **Message Integrity:** The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC), which is similar to HMAC

SSL Record Protocol

- SSL Record Protocol provides two services for SSL connection:
- **confidentiality**
 - using symmetric encryption with a shared secret key defined by Handshake Protocol
 - AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
 - message is compressed before encryption
- **message integrity**
 - using a MAC with shared secret key
 - similar to HMAC but with different padding

SSL Record Protocol Operation

- Takes an application message to be transmitted, fragments it into blocks, compresses the data, applies a MAC, encrypts, adds a header and transmits the unit in a TCP segment
- Received data are decrypted, verified, decompressed, and reassembled and then delivered to high-level user.

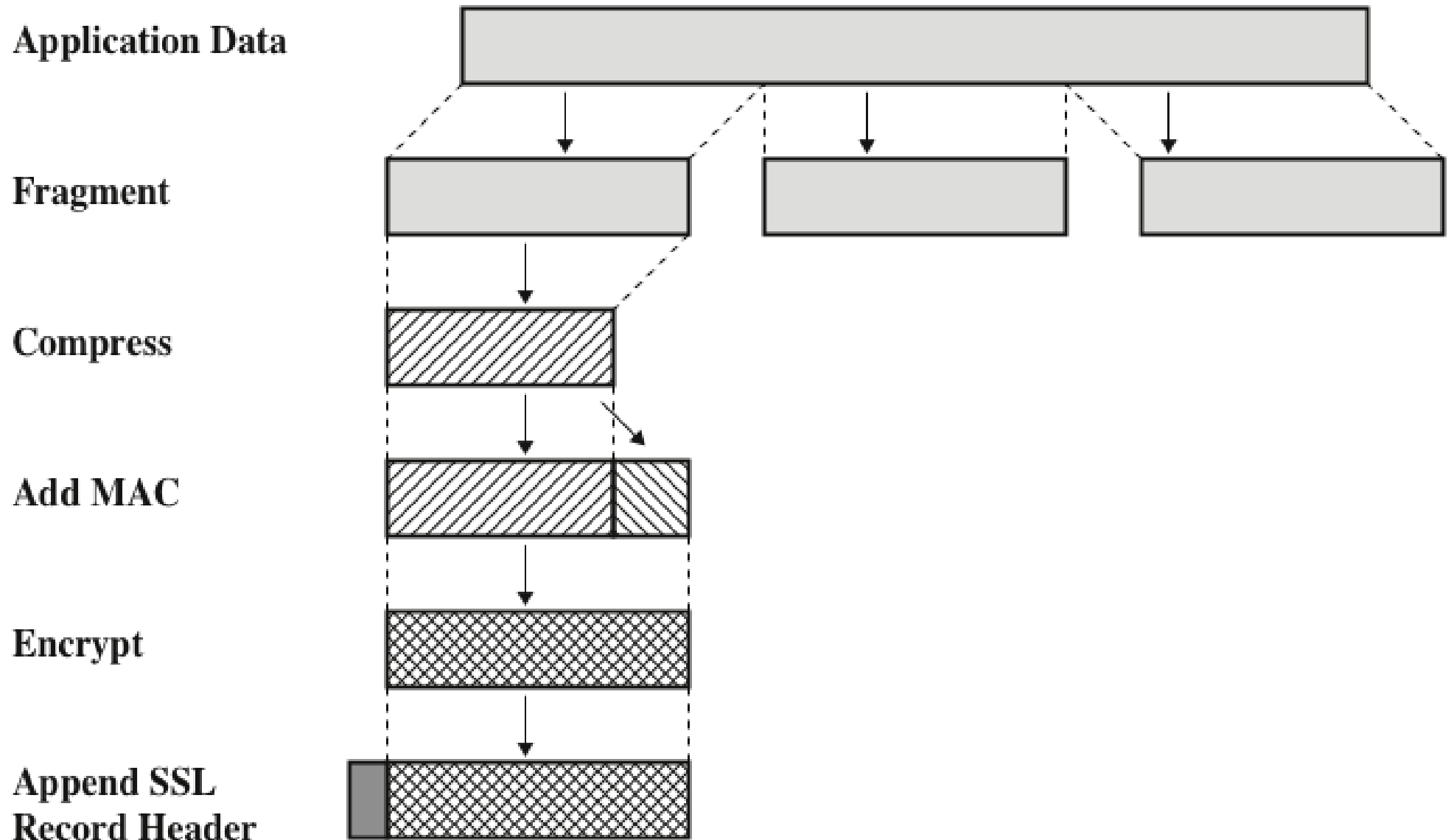
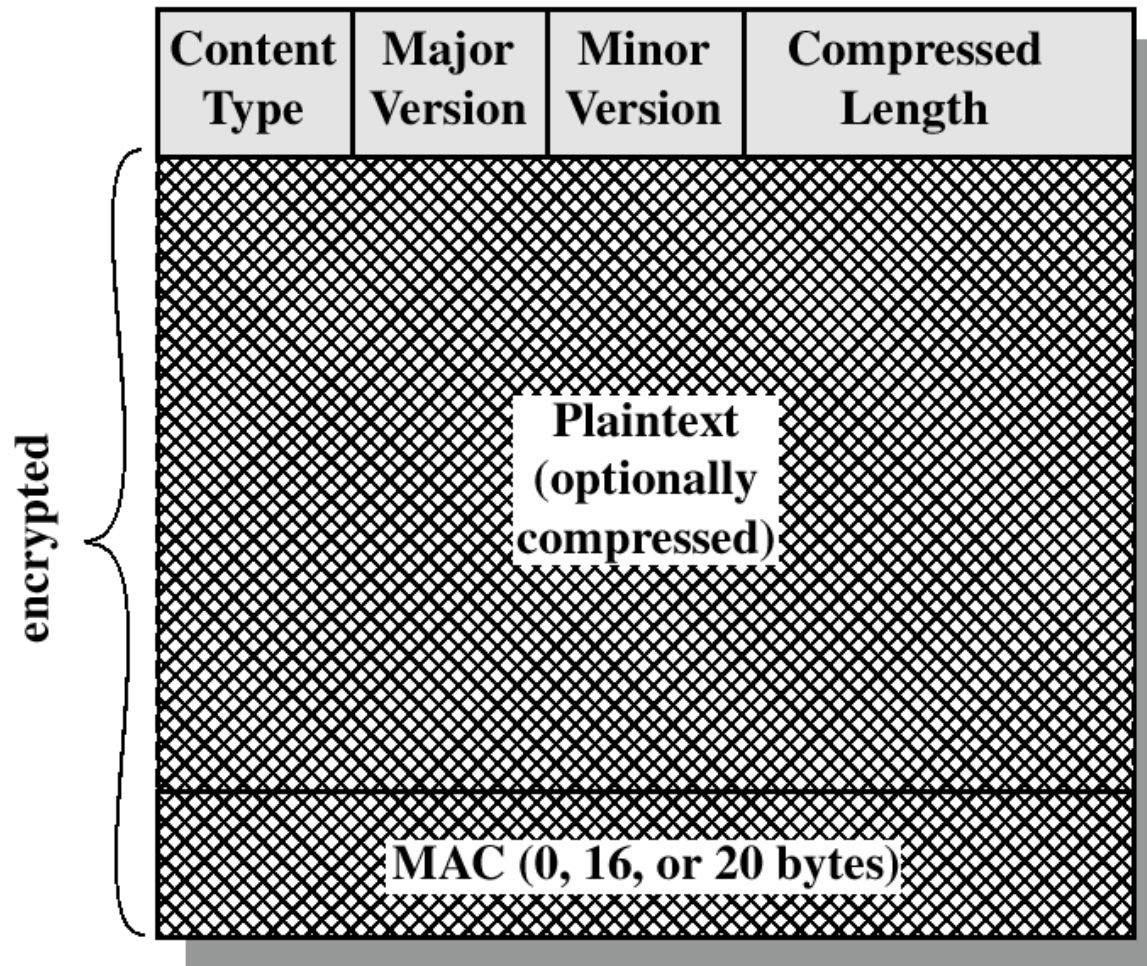


Figure 17.3 SSL Record Protocol Operation

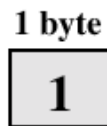
SSL Record Protocol Operation

- Fragmentation- message broken into 2^{14} byte blocks
- Compression - lossless and optional
- Compute a message authentication code (MAC) over the compressed data, using a shared secret key
- Compresses message and MAC are encrypted using symmetric encryption
- A header is prepared containing content type, versions and compressed length

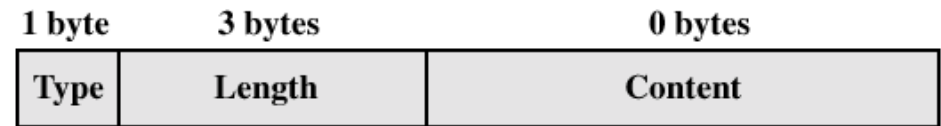
SSL Record Format



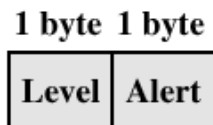
SSL Record Protocol Payload



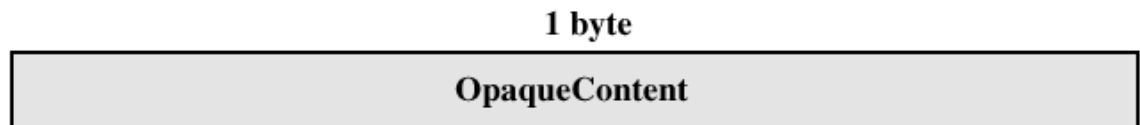
(a) Change Cipher Spec Protocol



(c) Handshake Protocol



(b) Alert Protocol



(d) Other Upper-Layer Protocol (e.g., HTTP)

SSL Change Cipher Spec Protocol

- One of 3 SSL specific protocols which use the SSL Record protocol-simplest
- A single message, single byte = 1
- Purpose causes pending state to become current
- Updates the cipher suite in use

1 byte



(a) Change Cipher Spec Protocol

SSL Alert Protocol

- Conveys SSL-related alerts to peer entity
- First byte - severity
 - warning or fatal (terminates)
- Second byte - specific alert
 - fatal: unexpected message, bad record MAC, decompression failure, handshake failure, illegal parameter
 - warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- compressed and encrypted like all SSL data

1 byte 1 byte

Level	Alert
-------	-------

(b) Alert Protocol

Handshake Protocol

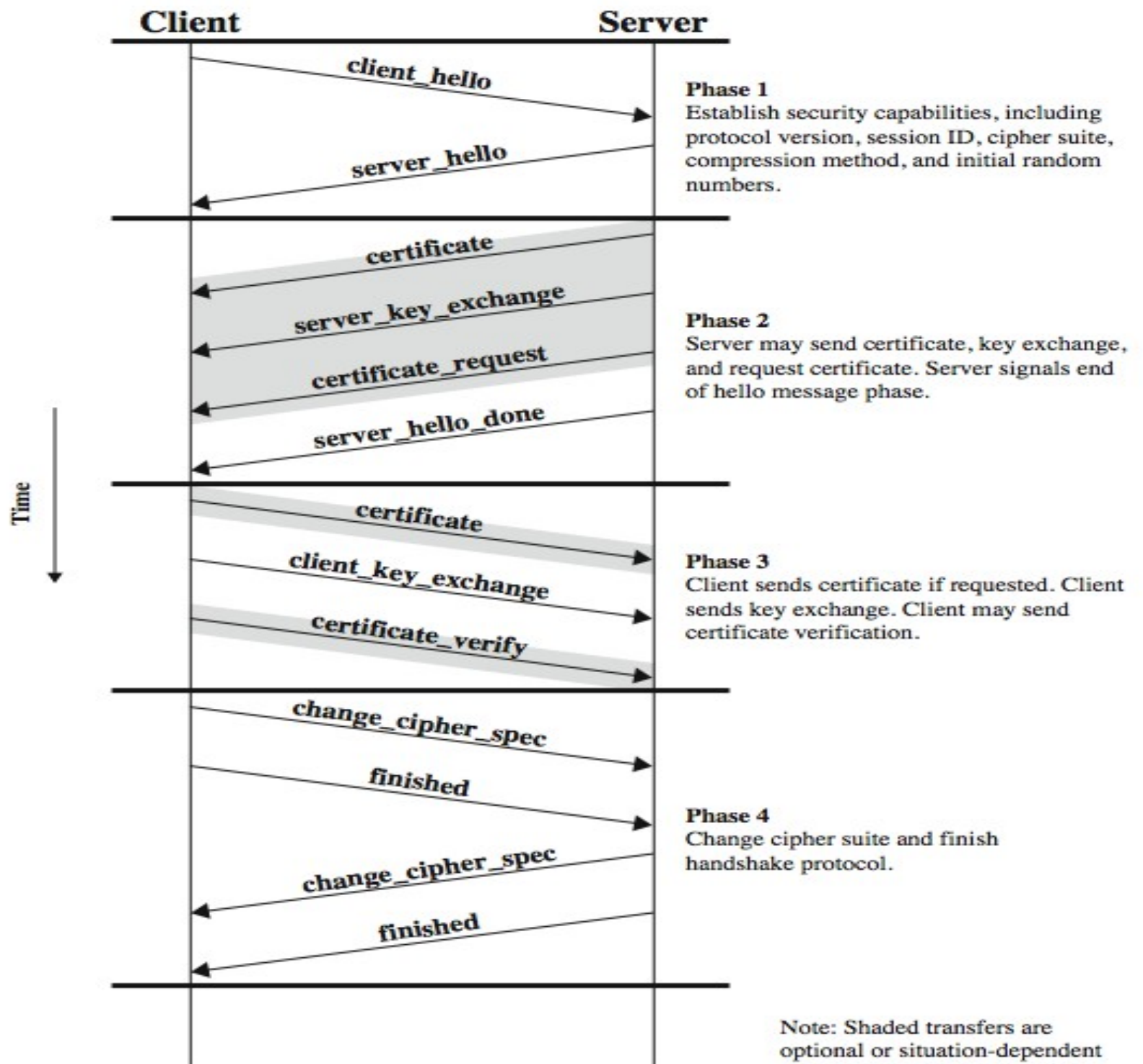
- Most complex part of SSL. Used before application data are transmitted.
- Allows the server and client:
 - to authenticate each other.
 - negotiate encryption, MAC algorithm and cryptographic keys.
- Comprises a series of messages in phases
 1. Establish Security Capabilities (RSA, DH...)
 2. Server Authentication and Key Exchange
 3. Client Authentication and Key Exchange
 4. Finish

1 byte	3 bytes	≥ 0 bytes
Type	Length	Content

(c) Handshake Protocol

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

Table SSL Handshake Protocol Message Types



Handshake Protocol Actions

Cryptographic Computations

- Two further items are of interest:
 - The creation of a shared master secret by means of the key exchange
 - The shared master secret is a one-time 48-byte value generated for this session by means of secure key exchange
 - The generation of cryptographic parameters from the master secret
 - CipherSpecs require a client write MAC secret, a server write MAC secret, a client write key, a server write key, a client write IV, and a server write IV which are generated from the master secret in that order
 - These parameters are generated from the master secret by hashing the master secret into a sequence of secure bytes of sufficient length for all needed parameters

Transport Layer Security (TLS)

- An IETF standardization initiative whose goal is to produce an Internet standard version of SSL
- Is defined as a Proposed Internet Standard in RFC 5246
 - RFC 5246 is very similar to SSLv3
 - Differences include:
 - Version number
 - Message Authentication Code
 - Pseudorandom function
 - Alert keys
 - Cipher suites
 - Client certificate types
 - Certificate_verify and Finished Messages
 - Cryptographic computations
 - Padding

Transport Layer Security

- IETF standard RFC 2246 similar to SSLv3
- Minor differences
 - in record format version number
 - uses HMAC for MAC
 - a pseudo-random function expands secrets
 - based on HMAC using SHA-1 or MD5
 - has additional alert codes
 - some changes in supported ciphers
 - changes in certificate types & negotiations
 - changes in crypto computations & padding

Let us Recall HMAC

<http://en.wikipedia.org/wiki/HMAC>

Then HMAC(K,m) is mathematically defined by

$$\text{HMAC}(K,m) = H((K \oplus \text{opad}) \parallel H((K \oplus \text{ipad}) \parallel m))$$

- $H(\cdot)$ be a cryptographic hash function
- K be a secret key padded to the right with extra zeros to the block size of the hash function
- m be the message to be authenticated
- \parallel denote concatenation and \oplus denote exclusive or (XOR)
- **opad** be the outer padding (**0x5c5c5c...5c5c**, one-block-long hexadecimal constant)
- **ipad** be the inner padding (**0x363636...3636**, one-block-long hexadecimal constant)

HMAC

➤ specified as Internet standard RFC2104

➤ uses hash function on the message:

$$\text{HMAC}_K(M) = \text{Hash}[(K^+ \text{ XOR opad}) \parallel \text{Hash}[(K^+ \text{ XOR ipad}) \parallel M]]$$

- where K^+ is the key padded out to size
 - opad, ipad are specified padding constants
- overhead is just 3 more hash calculations than the message needs alone
- any hash function can be used
- eg. MD5, SHA-1, RIPEMD-160, Whirlpool

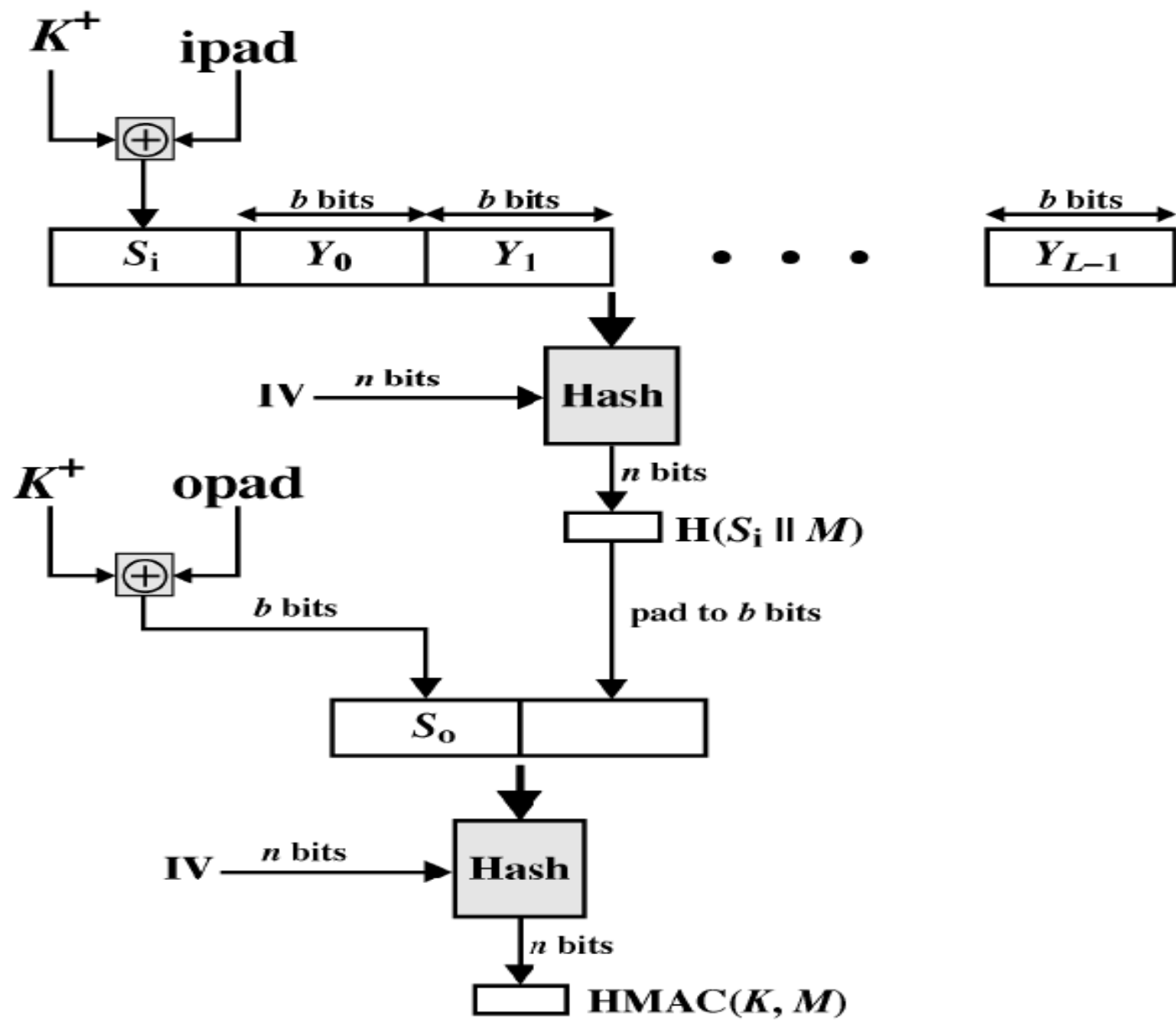


Figure 3.6 HMAC Structure

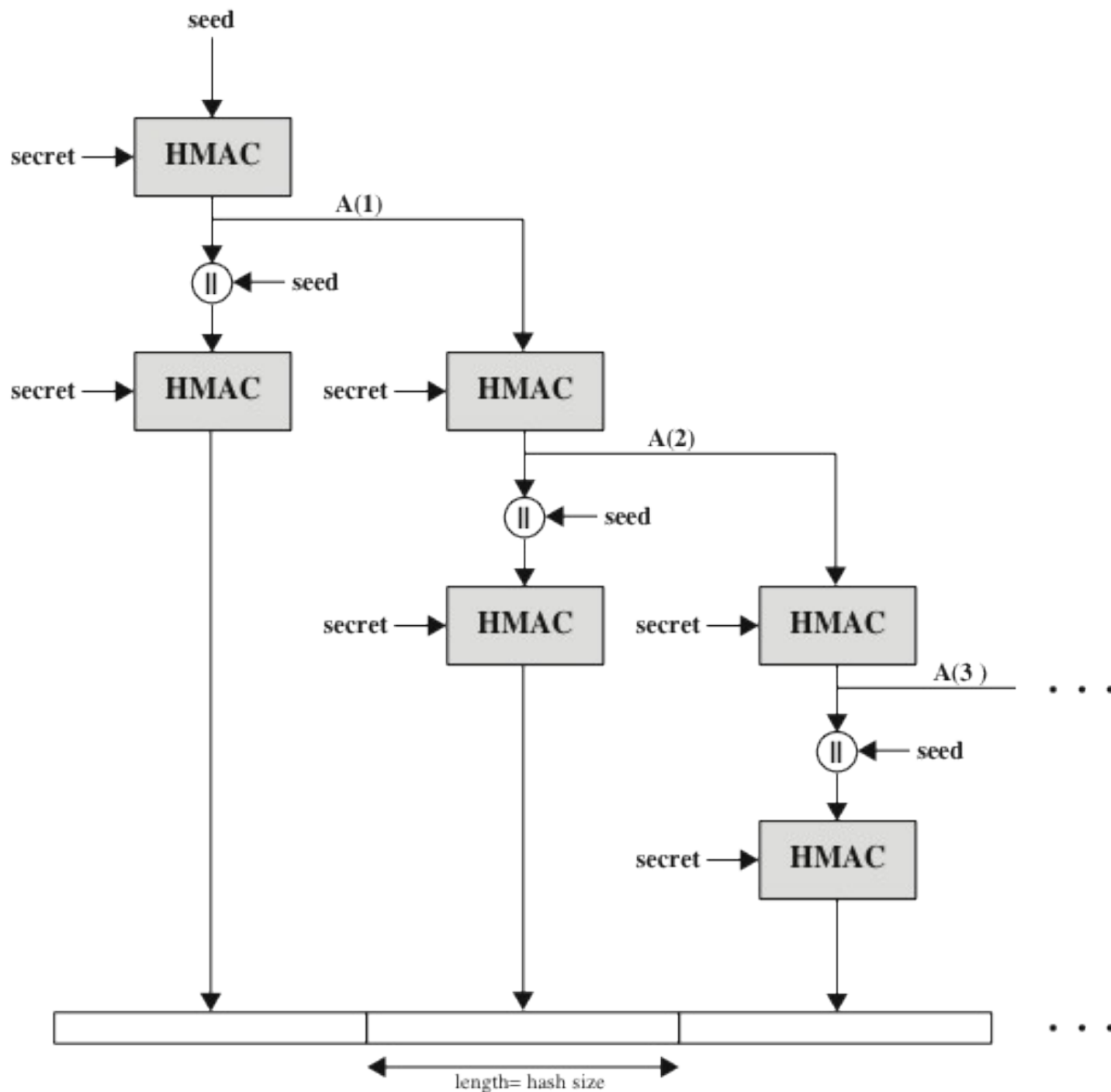


Figure 17.7 TLS Function P_hash (secret, seed)

HTTPS

(HTTP over SSL)

- Refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server
- The HTTPS capability is built into all modern Web browsers
- A user of a Web browser will see URL addresses that begin with https:// rather than http://
- If HTTPS is specified, port 443 is used, which invokes SSL
- Documented in RFC 2818, *HTTP Over TLS*
 - There is no fundamental change in using HTTP over either SSL or TLS and both implementations are referred to as HTTPS
- When HTTPS is used, the following elements of the communication are encrypted:
 - URL of the requested document
 - Contents of the document
 - Contents of browser forms
 - Cookies sent from browser to server and from server to browser
 - Contents of HTTP header

Connection Initiation

For HTTPS, the agent acting as the HTTP client also acts as the TLS client. The client initiates a connection to the server on the appropriate port and then sends the TLS ClientHello to begin the TLS handshake.

When the TLS handshake has finished, the client may then initiate the first HTTP request.

All HTTP data is to be sent as TLS application data. Normal HTTP behavior, including retained connections, should be followed.

There are three levels of awareness of a connection in HTTPS. At the HTTP level, an HTTP client requests a connection to an HTTP server by sending a connection request to the next lowest layer. Typically, the next lowest layer is TCP, but it also may be TLS/SSL.

At the level of TLS, a session is established between a TLS client and a TLS server. This session can support one or more connections at any time. As we have seen, a TLS request to establish a connection begins with the establishment of a TCP connection between the TCP entity on the client side and the TCP entity on the server side.

Connection Closure

- An HTTP client or server can indicate the closing of a connection by including the line `Connection: close` in an HTTP record
- The closure of an HTTPS connection requires that TLS close the connection with the peer TLS entity on the remote side, which will involve closing the underlying TCP connection
- TLS implementations must initiate an exchange of closure alerts before closing a connection
 - A TLS implementation may, after sending a closure alert, close the connection without waiting for the peer to send its closure alert, generating an “incomplete close”
- An unannounced TCP closure could be evidence of some sort of attack so the HTTPS client should issue some sort of security warning when this occurs

Security of SSL and TLS

Reference: http://en.wikipedia.org/wiki/Transport_Layer_Security#Security

- SSL 2.0 is flawed in a variety of ways
 - Identical cryptographic keys are used for message authentication and encryption.
 - SSL 2.0 has a weak MAC construction that uses the MD5 hash function with a secret prefix, making it vulnerable to length extension attacks.
 - SSL 2.0 does not have any protection for the handshake, meaning a man-in-the-middle downgrade attack can go undetected.
 - SSL 2.0 uses the TCP connection close to indicate the end of data. This means that truncation attacks are possible: the attacker simply forges a TCP FIN, leaving the recipient unaware of an illegitimate end of data message (SSL 3.0 fixes this problem by having an explicit closure alert).
 - SSL 2.0 assumes a single service and a fixed domain certificate, which clashes with the standard feature of virtual hosting in Web servers. This means that most websites are practically impaired from using SSL.
- SSL 2.0 is disabled by default, beginning with Internet Explorer 7, Mozilla Firefox 2 Opera 9.5 and Safari 1.5

Security of SSL and TLS

- SSL 3.0 improved upon SSL 2.0 by adding SHA-1 based ciphers and support for certificate authentication.
- From a security standpoint, SSL 3.0 should be considered less desirable than TLS 1.0.
- The SSL 3.0 cipher suites have a weaker key derivation process; half of the master key that is established is fully dependent on the MD5 hash function, which is not resistant to collisions and is, therefore, not considered secure.

Under TLS 1.0, the master key that is established depends on both MD5 and SHA-1 so its derivation process is not currently considered weak.

- It is for this reason that SSL 3.0 implementations cannot be validated under FIPS 140-2

Security of TLS

TLS has a variety of security measures:

- Protection against a downgrade of the protocol to a previous (less secure) version or a weaker cipher suite.
- Numbering subsequent Application records with a sequence number and using this sequence number in the message authentication codes (MACs).
- Using a message digest enhanced with a key (so only a key-holder can check the MAC). The HMAC construction used by most TLS cipher suites is specified in RFC 2104 (SSL 3.0 used a different hash-based MAC).
- The message that ends the handshake ("Finished") sends a hash of all the exchanged handshake messages seen by both parties.
- The pseudorandom function splits the input data in half and processes each one with a different hashing algorithm (MD5 and SHA-1), then XORs them together to create the MAC. This provides protection even if one of these algorithms is found to be vulnerable.

Attacks against TLS/SSL

- **Renegotiation attack**

- it allows an attacker who can hijack an https connection to splice their own requests into the beginning of the conversation the client has with the web server.
- The attacker can't actually decrypt the client-server communication, so it is different from a typical man-in-the-middle attack. A short-term fix is for web servers to stop allowing renegotiation, which typically will not require other changes unless client certificate authentication is used.
- To fix the vulnerability, a renegotiation indication extension was proposed for TLS.

Attacks against TLS/SSL

- Version rollback attacks
- BEAST attack (Browser Exploit Against SSL/TLS) using a Java applet to violate same origin policy constraints, for a long-known cipher block chaining (CBC) vulnerability in TLS 1.0

Practical exploits had not been previously demonstrated for this vulnerability, which was originally in 2002.

The vulnerability of the attack had been fixed with TLS 1.1 in 2006

- RC4 attacks

In spite of existing attacks on RC4 that break it, the cipher suites based on RC4 in SSL and TLS were considered secure because of how the cipher was used in these protocols. In 2011 RC4 suite was actually recommended as a work around for the BEAST attack. But, newly statistical biases in RC4 key table were discovered to recover parts of plaintext with large number of TLS encryptions.

Attacks against TLS/SSL

- **Truncation attack** Published in July 2013, the attack causes web services such as Gmail and Hotmail to display a page that informs the user that they have successfully signed-out, while ensuring that the user's browser maintains authorization with the service, allowing an attacker with subsequent access to the browser to access and take over control of the user's logged-in account.
- Excellent presentation titled “**Cryptographic Analysis of TLS**” given by Kenny Paterson for TLS Security (in FOSAD 2013):

<http://www.sti.uniurb.it/events/fosad13/slides/paterson-fosad13.pdf>

See Slide 167 for some discussion of the use of TLS

(“Most TLS implementations now patched against BEAST; Many TLS implementations patched against Lucky 13 ; No simple TLS patch for RC4 attack; Disable TLS compression to prevent CRIME;

We need TLS 1.2!”)

Attacks against TLS/SSL

- **Apple's SSL/TLS bug (22 Feb 2014)**

<https://www.imperialviolet.org/2014/02/22/applebug.html>

<http://www.imore.com/understanding-apples-ssl-tls-bug>

“ The result of this code is that an attacker on the same network as you could perform a man-in-the-middle attack where they fake a certificate keychain to a secure site, like your bank. You can't trust any secured connections in affected version of iOS and OS X. Everybody should update their iOS devices and Apple TVs if they haven't already.”

Attacks against TLS/SSL

- **Heartbleed Bug**

The Heartbleed bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

- The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

Attacks against TLS/SSL

http://en.wikipedia.org/wiki/Transport_Layer_Security#Security

Survey of websites [\[edit\]](#)

As of April 2014, Trustworthy Internet Movement estimate the ratio of websites that are vulnerable to TLS attacks.^[14]

Survey of the TLS vulnerabilities of the most popular websites

Attacks	Security			
	Insecure	Depends	Secure	Other
Renegotiation attack	5.5% (−0.2%) support insecure renegotiation	1.4% (−0.9%) support both	85.9% (+1.2%) support secure renegotiation	7.2% (−0.1%) not support
RC4 attacks	33.4% (±0.0%) support RC4 suites used with modern browsers	57.4% (−0.6%) support some RC4 suites	9.3% (+0.6%) not support	N/A
BEAST attack	71.8% (±0.0%) vulnerable	N/A	N/A	N/A
CRIME attack	11.4% (−1.5%) vulnerable	N/A	N/A	N/A
Heartbleed	0.8% (−) vulnerable	N/A	N/A	N/A

Security of Javascript

- “ JavaScript and the DOM provide the potential for malicious authors to deliver scripts to run on a client computer via the web. Browser authors contain this risk using two restrictions.
- First, scripts run in a sandbox in which they can only perform web-related actions, not general-purpose programming tasks like creating files. Second, scripts are constrained by the same origin policy: scripts from one web site do not have access to information such as usernames, passwords, or cookies sent to another site. Most JavaScript-related security bugs are breaches of either the same origin policy or the sandbox.
- There are subsets of general JavaScript — ADsafe, Secure ECMA Script (SES) — that provide greater level of security, especially on code created by third parties ... “

<http://en.wikipedia.org/wiki/JavaScript#Security>

- For the detailed discussion on Javascript security, see another presentation titled “Information flow control for the web” by Prof. Frank PIESENS in FOSAD 2013:
<http://www.sti.uniurb.it/events/fosad13/slides/piessens-fosad13.pdf>
- Recommended Video: JavaScript Security
<http://www.youtube.com/watch?v=hzAf5PY2Ws0>

Example of Cryptographic Misuse

- “An Empirical Study of Cryptographic Misuse in Android Applications”

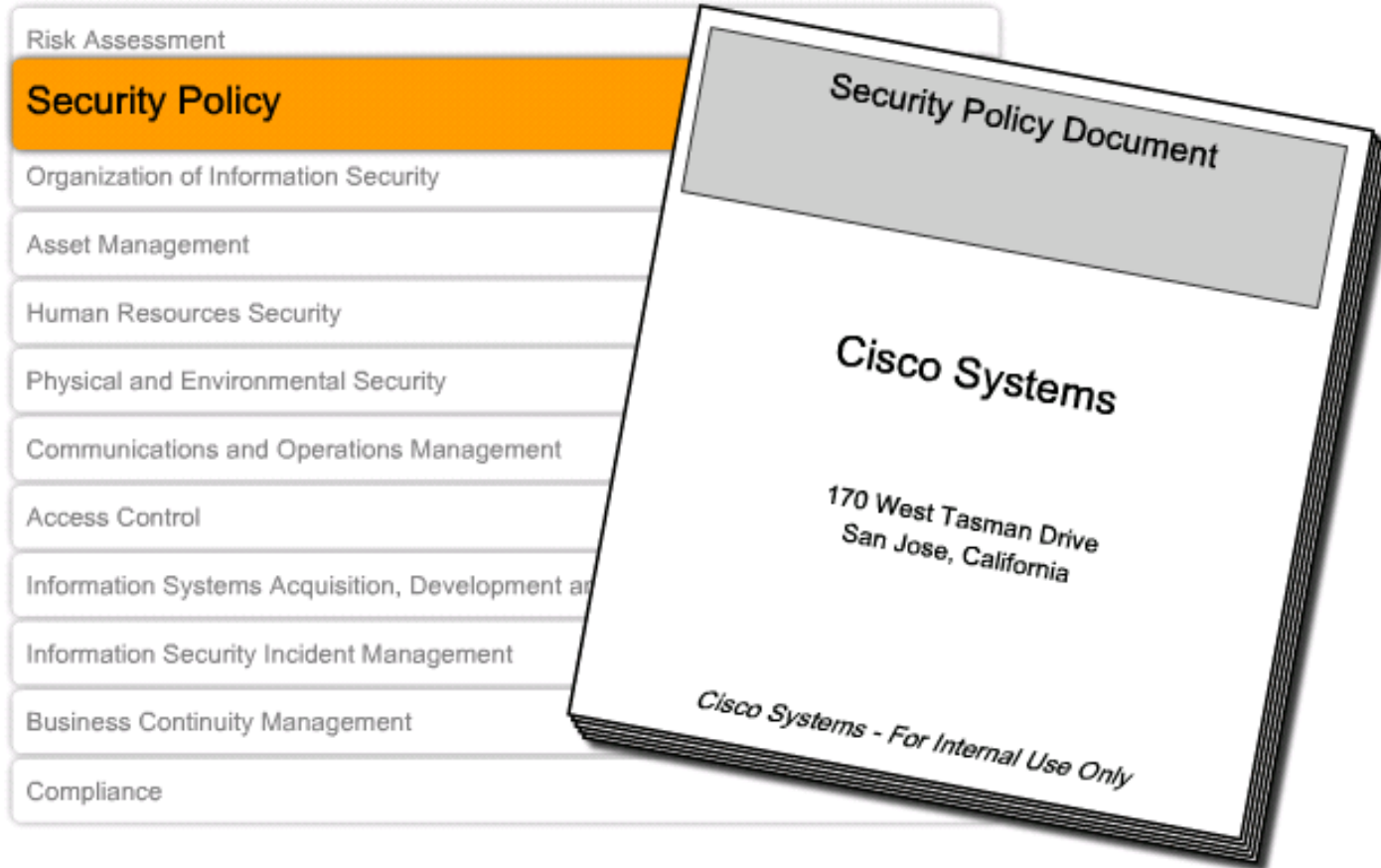
www.cs.ucsb.edu/~chris/research/doc/ccs13_cryptolint.pdf

- “ ..We develop program analysis techniques to automatically check programs on the Google Play marketplace, and find that 10,327 out of 11,748 applications that use cryptographic APIs 88% overall – make at least one mistake..”
- **Some rules violated:**
 - Rule 1: Do not use ECB mode for encryption
 - Rule 2: Do not use a non-random IV for CBC encryption
 - Rule 3: Do not use constant encryption keys

Domains of Network Security

- It is also important to have an understanding of the various network security domains.
 - Domains provide an organized framework to facilitate learning about network security.
- ISO/IEC 27002 specifies 12 network security domains.
 - These 12 domains serve to organize at a high level the vast realm of information under the umbrella of network security.
 - The 12 domains are intended to serve as a common basis for developing organizational security standards and effective security management practices, and to help build confidence in inter-organizational activities.

Security Policy



Types of Attacks

- There are four categories of attacks:
 - Malicious Code: Viruses, Worms and Trojan Horses
 - Reconnaissance Attacks
 - Access Attacks
 - Denial of Service (DoS) Attacks

Let's focus on Malicious Code

Malware

- “Malicious software” is software designed to infiltrate a computer without the owner's informed consent.
- Malware includes:
 - Computer viruses
 - Worms
 - Trojan horses
 - Rootkits
 - Backdoors (Method of bypassing normal authentication procedures and usually installed using Trojan horses or worms.)
 - For profit (Spyware, botnets, keystroke loggers, and dialers)



Spyware

- Spyware is a strictly for-profit category of malware designed to:
 - Monitor a users web browsing.
 - Display unsolicited advertisements.
 - Redirect affiliate marketing revenues to the spyware creator.
- Spyware programs are generally installed by exploiting security holes or as Trojan horse programs such as most peer-to-peer applications.

Why Write Malicious Code?

- Most early worms and viruses were written as experiments or pranks generally intended to be harmless or merely annoying rather than to cause serious damage to computers.
- Young programmers learning about viruses and the techniques wrote them for the sole purpose that they could or to see how far it could spread.
 - In some cases the perpetrator did not realize how much harm their creations could do.
- As late as 1999, widespread viruses such as the Melissa virus appear to have been written chiefly as pranks.

Malicious Code Writing Today

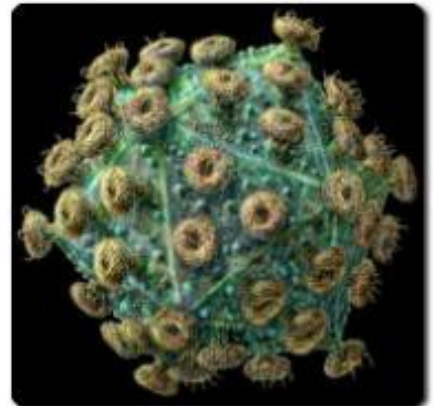
- Malicious code writing has changed for profitable reasons.
 - Mainly due to the Internet and broadband access.
 - Since 2003 the majority of viruses and worms have been designed to take control of users' computers for black-market exploitation.
 - Infected "zombie computers" are used to send email spam, to host contraband data, or to engage in DDoS attacks as a form of extortion.
- In 2008, Symantec published:
 - The release rate of malicious code and other unwanted programs may be exceeding that of legitimate software applications.

Viruses, Trojan horses, and Worms

- A **virus** is malicious software that is attached to another program to execute a particular unwanted function on a user's workstation.
- A **worm** executes arbitrary code and installs copies of itself in the infected computer's memory, which infects other hosts.
- A **Trojan** horse is different only in that the entire application was written to look like something else, when in fact it is an attack tool.

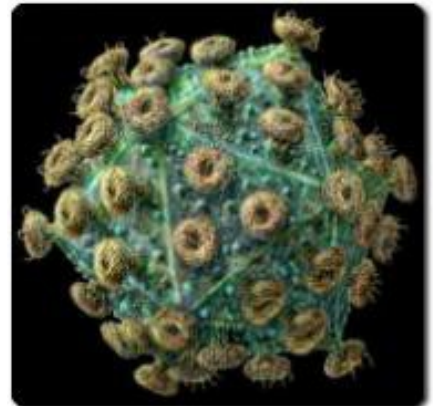
Viruses

- A computer virus is a malicious computer program (executable file) that can copy itself and infect a computer without permission or knowledge of the user.
- A virus can only spread from one computer to another by:
 - Sending it over a network as a file or as an email payload.
 - Carrying it on a removable medium.
- Viruses need **USER INTERVENTION** to spread ...



Viruses

- A computer virus is a malicious computer program (executable file) that can copy itself and infect a computer without permission or knowledge of the user.
- A virus can only spread from one computer to another by:
 - Sending it over a network as a file or as an email payload.
 - Carrying it on a removable medium.
- Viruses need **USER INTERVENTION** to spread ...

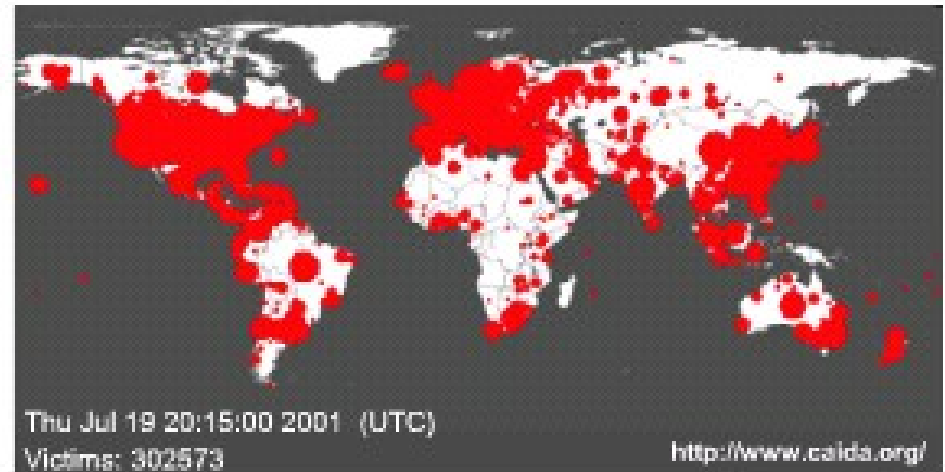
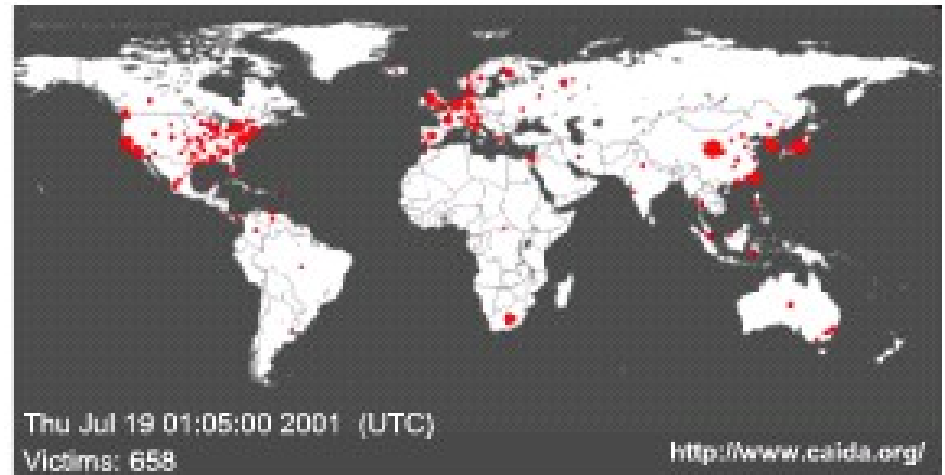


Worms

- Worms are a particularly dangerous type of hostile code.
 - They replicate themselves by independently exploiting vulnerabilities in networks.
 - Worms usually slow down networks.
- Worms DO NOT NEED USER INTERVENTION!
 - Worms do not require user participation and can spread extremely fast over the network.

SQL Slammer Worms

- In January 2001, the SQL Slammer Worm slowed down global Internet traffic as a result of DoS.
- Over 250,000 hosts were affected within 30 minutes of its release.
- The worm exploited a buffer overflow bug in Microsoft's SQL Server.
 - A patch for this vulnerability was released in mid-2002, so the servers that were affected were those that did not have the update patch applied.



Anatomy of a Worm

- The enabling vulnerability
 - A worm installs itself using an exploit vector on a vulnerable system.
- Propagation mechanism
 - After gaining access to devices, a worm replicates and selects new targets.
- Payload
 - Once the device is infected with a worm, the attacker has access to the host – often as a privileged user.
 - Attackers could use a local exploit to escalate their privilege level to administrator.

The year's most-hacked software 2009

- *“Kits that go by names like ‘T-IFramer,’ ‘Liberty Exploit Systems’ and ‘Elenore’ all turned up on underground markets selling for \$300 to \$500, Kandek says, and allow the attacker to install a Trojan program ready to download whatever malicious software a cybercriminal wishes, from spyware to click-fraud software. All three of those kits exploit three unique Adobe Reader bugs, along with a smaller number of bugs in Internet Explorer, Microsoft Office, Firefox and even Quicktime.”*

Excerpt from the article at:

<http://www.cbc.ca/technology/story/2009/12/16/f-forbes-adobe-hacked-software.html>

Trojan Horse

- A Trojan horse is a program that appears, to the user, to perform a desirable function but, in fact, facilitates unauthorized access to the user's computer system.
- Trojan horses may appear to be useful or interesting programs, or at the very least harmless to an unsuspecting user, but are actually harmful when executed.
- Trojan horses are not self-replicating which distinguishes them from viruses and worms.



Trojan Horse Classification

- Remote-access Trojan Horse
 - Enables unauthorized remote access
- Data sending Trojan Horse
 - Provides the attacker with sensitive data such as passwords
- Destructive Trojan Horse
 - Corrupts or deletes files
- Proxy Trojan Horse
 - User's computer functions as a proxy server
- FTP Trojan Horse (opens port 21)
 - Security software disabler Trojan Horse (stops anti-virus programs or firewalls from functioning)
- Denial of Service Trojan Horse (slows or halts network activity)



Five Phases of a Virus/Worm Attack

- Probe phase:
 - Vulnerable targets are identified using ping scans.
 - Application scans are used to identify operating systems and vulnerable software.
 - Hackers obtain passwords using social engineering, dictionary attack, brute-force, or network sniffing.
- Penetrate phase:
 - Exploit code is transferred to the vulnerable target.
 - Goal is to get the target to execute the exploit code through an attack vector, such as a buffer overflow, ActiveX or Common Gateway Interface (CGI) vulnerabilities, or an email virus.
- Persist phase:
 - After the attack is successfully launched in the memory, the code tries to persist on the target system.
 - Goal is to ensure that the attacker code is running and available to the attacker even if the system reboots.
 - Achieved by modifying system files, making registry changes, and installing new code.
- Propagate phase:
 - The attacker attempts to extend the attack to other targets by looking for vulnerable neighboring machines.
 - Propagation vectors include emailing copies of the attack to other systems, uploading files to other systems using file shares or FTP services, active web connections, and file transfers through Internet Relay Chat.
- Paralyze phase:
 - Actual damage is done to the system.
 - Files can be erased, systems can crash, information can be stolen, and distributed DDoS attacks can be launched.

Exploit Comparison

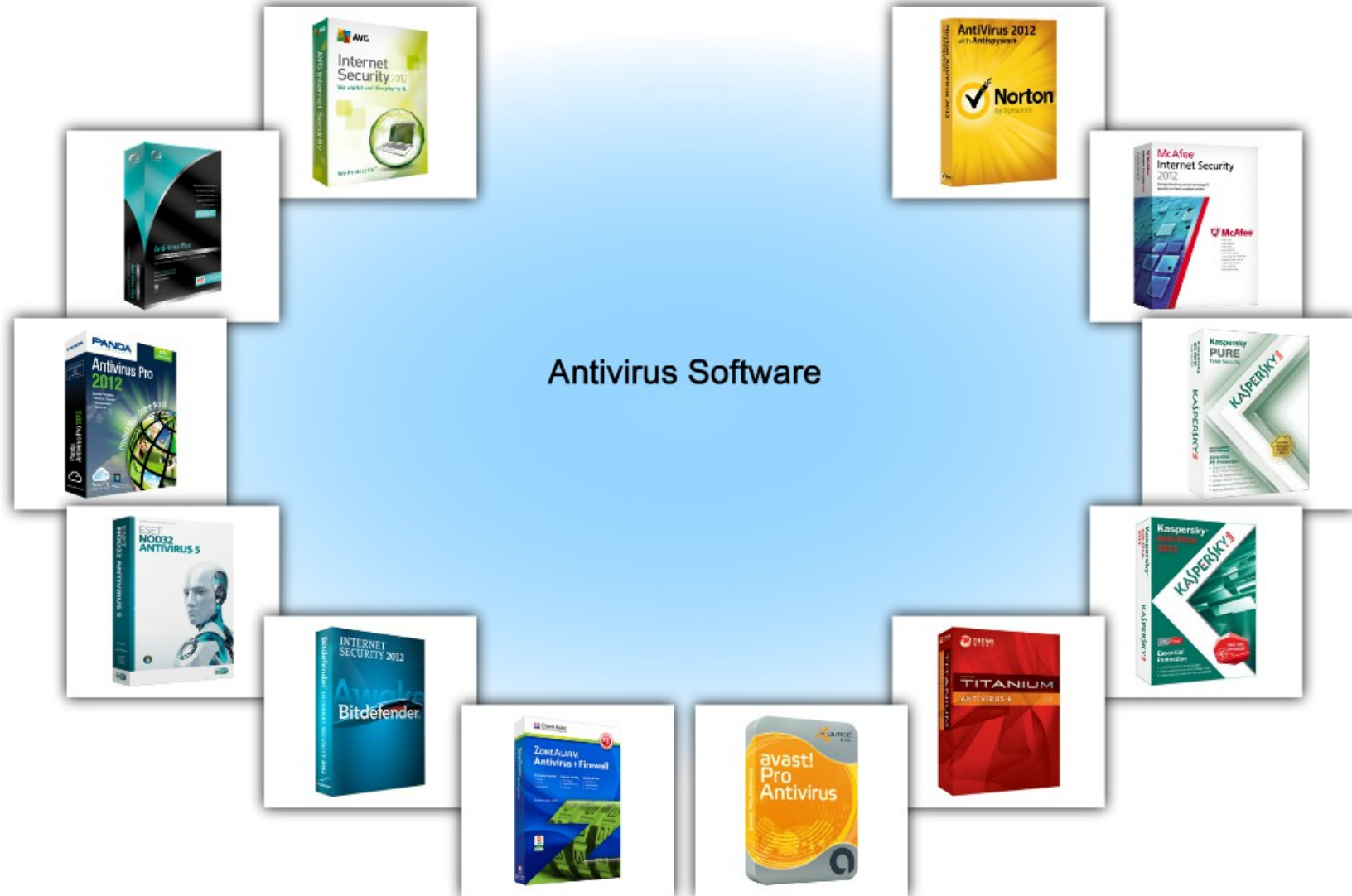
Worm and Virus – Exploit Comparison (~20 Yrs)

	Morris 1988	Love Bug 2000	Code Red 2001	Slammer 2003	MyDoom 2004	Zotob 2005	MS RPC DNS Oday 2007
Probe	Scans for finger	N/A	Scans for IIS	N/A	N/A	Scans for MS directory services	Scans for endpoint Mapper query
Penetrate	Causes buffer overflow in fingerd	Arrives as email attachment	Causes buffer overflow in IIS	Causes buffer overflow in SQL and MSDE	Arrives as email attachment	Causes buffer overflow in UPnP service	Causes buffer overflow in RPC service
Persist	Executes script to download code	Creates executables and edits the registry	Executes script to download code	N/A	Creates executables and edits the registry	Creates executables and edits the registry, download code	Executes payload to download code
Propagate	Looks for addresses and spreads to new victims	Opens address book and emails copies of itself to new victims	Picks new addresses and spreads to new victims	Picks new addresses and spreads to new victims	Opens address book and email copies of itself to new victims	Starts FTP and TFTP services, looks for addresses and spreads to new victims	Looks for addresses and spreads to new victims
Paralyze	Spawns many processes which slow the system	Worm spreads	Spawns many threads which slow the system	Generates many packets which slows the network	Worm spreads	Deletes registry keys and files, and terminates processes	Worm spreads

Commonalities

- A majority of the software vulnerabilities that are discovered relate to buffer overflows.
 - Buffer overflows are usually the primary conduit through which viruses, worms, and Trojan Horses do their damage.
- Viruses and Trojan Horses tend to take advantage of local root buffer overflows.
 - A root buffer overflow is intended to attain root privileges to a system.
- Worms such as SQL Slammer and Code Red exploit remote root buffer overflows.
 - Remote root buffer overflows are similar to local root buffer overflows, except that local end user or system intervention is not required.

How Do You Mitigate Viruses and Worms?

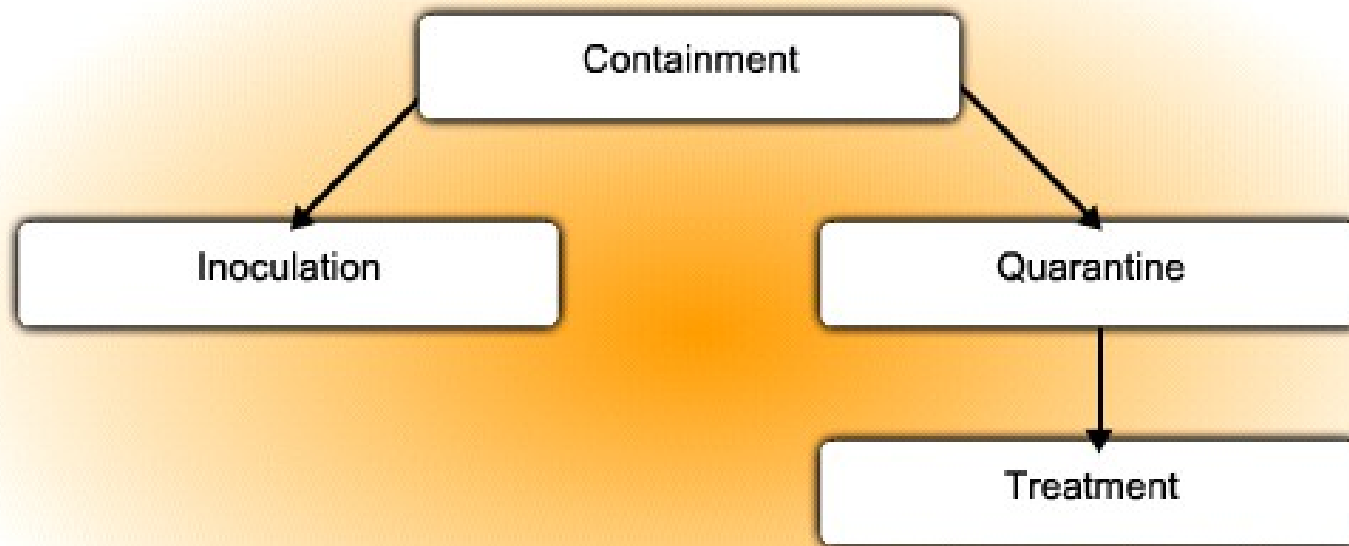


Viruses and Trojan Horses - Mitigation

- The primary means of mitigating virus and Trojan horse attacks is anti-virus software.
 - For total protection, host-based intrusion prevention systems (HIPS), such as Cisco Security Agent should also be deployed.
 - HIPS protects the OS kernel.
- Anti-virus software helps prevent hosts from getting infected and spreading malicious code.
 - However, antivirus software must be used properly.
 - Always update with the latest antivirus .dat and application versions.
 - Consider that it requires much more time to clean up infected computers than it does to maintain up-to-date anti-virus software and anti-virus definitions on the same machines.

Mitigating an Active Worm

- Worm attack mitigation requires diligence on the part of system and network administration staff.
- There is a four phase process to mitigate an active worm attacks.



Worms - Mitigation

- Containment Phase:
 - Limit the spread of a worm infection to areas of the network that are already affected.
 - Compartmentalize and segment the network to slow down or stop the worm to prevent currently infected hosts from targeting and infecting other systems.
 - Use both outgoing and incoming ACLs on routers and firewalls at control points within the network.
- Inoculation Phase:
 - Runs parallel to or subsequent to the containment phase.
 - All uninfected systems are patched with the appropriate vendor patch for the vulnerability.
 - The inoculation process further deprives the worm of any available targets.

Worms - Mitigation

- Containment Phase:
 - Limit the spread of a worm infection to areas of the network that are already affected.
 - Compartmentalize and segment the network to slow down or stop the worm to prevent currently infected hosts from targeting and infecting other systems.
 - Use both outgoing and incoming ACLs on routers and firewalls at control points within the network.
- Inoculation Phase:
 - Runs parallel to or subsequent to the containment phase.
 - All uninfected systems are patched with the appropriate vendor patch for the vulnerability.
 - The inoculation process further deprives the worm of any available targets.

Example: Mitigating SQL Slammer

- The SQL Slammer worm used UDP port 1434.
 - This port should normally be blocked by a firewall on the perimeter.
 - However, most infections enter internally and therefore, to prevent the spreading of this worm it would be necessary to block this port on all devices throughout the internal network.
- When SQL Slammer was propagating, some organizations could not block UDP port 1434 because it was required to access the SQL Server for legitimate business transactions.
 - Permit only selective access to a small number of clients using SQL Server.

Types of Attacks

- There are four categories of attacks:
 - Malicious Code: Viruses, Worms and Trojan Horses
 - Reconnaissance Attacks
 - Access Attacks
 - Denial of Service (DoS) Attacks

Let's focus on Malicious Code

Reconnaissance

- Reconnaissance also known as information gathering is the unauthorized discovery and mapping of systems, services, or vulnerabilities.
 - In most cases, precedes an access or DoS attack.
- Reconnaissance attacks can consist of the following:
 - Internet information queries
 - Ping sweeps
 - Port scans
 - Packet sniffers

Internet Information Queries

- DNS queries can reveal information such as who owns a particular domain and what addresses have been assigned to that domain.
 - Use tools such as **whois**, **nslookup**, ...

Ping Sweeps and Port Scans

- A ping sweep, or ICMP sweep, scans to determine which range of IP addresses map to live hosts.
- A port scan consists of sending a message to each port, one port at a time.
 - Response received indicates whether the port is used and can therefore be probed for weakness.

Packet Sniffing

- A packet sniffer is a software application that uses a network adapter card in promiscuous mode to capture all network packets that are sent across a LAN.
 - Packet sniffers can only work in the same collision domain as the network being attacked.
 - Promiscuous mode is a mode in which the network adapter card sends all packets that are received on the physical network wire to an application for processing.
 - Wireshark is an example of a packet sniffer.

Packet Sniffing

- A packet sniffer is a software application that uses a network adapter card in promiscuous mode to capture all network packets that are sent across a LAN.
 - Packet sniffers can only work in the same collision domain as the network being attacked.
 - Promiscuous mode is a mode in which the network adapter card sends all packets that are received on the physical network wire to an application for processing.
 - Wireshark is an example of a packet sniffer.

Types of Attacks

- There are four categories of attacks:
 - Malicious Code: Viruses, Worms and Trojan Horses
 - Reconnaissance Attacks
 - Access Attacks
 - Denial of Service (DoS) Attacks

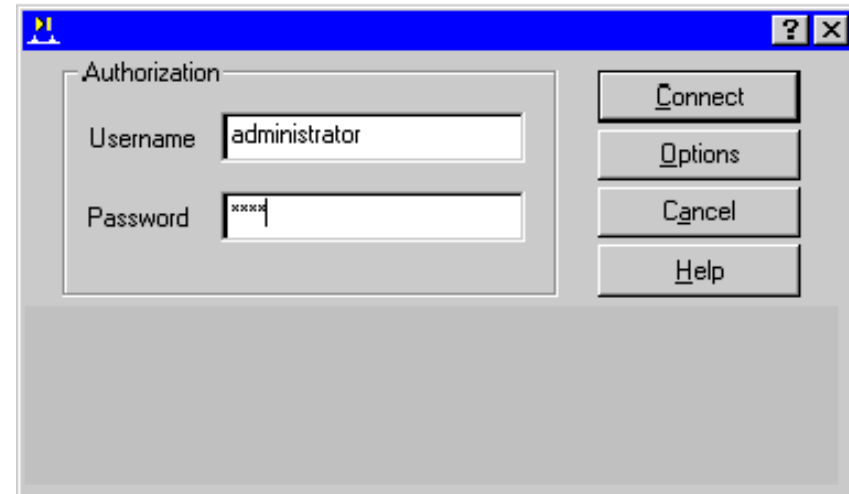
Let's focus on Malicious Code

Access Attacks

- Access attacks exploit known vulnerabilities in authentication services, FTP services, and web services to gain entry to web accounts, confidential databases, and other sensitive information for these reasons:
 - Retrieve data
 - Gain access
 - Escalate their access privileges

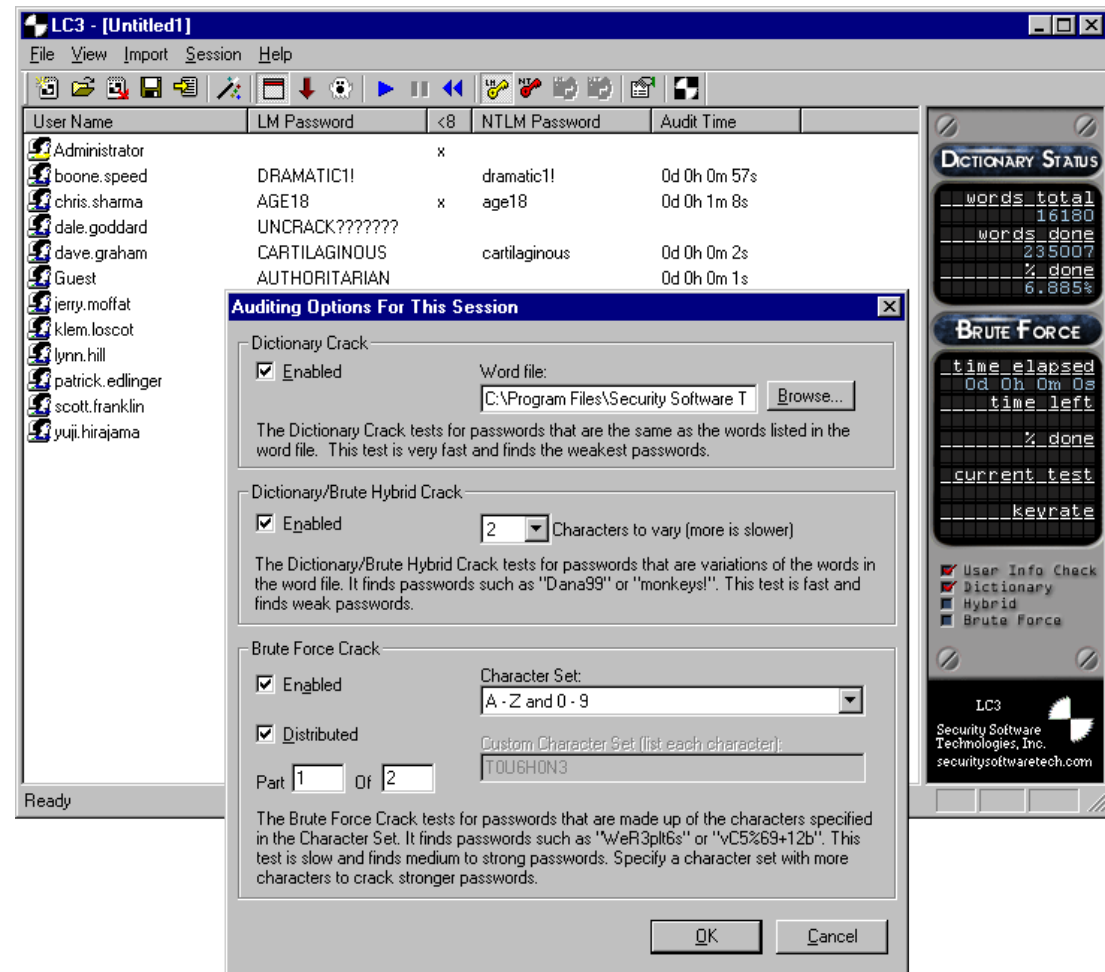
Password Attacks

- Hackers implement password attacks using the following:
 - Brute-force attacks
 - Trojan horse programs
 - IP spoofing
 - Packet sniffers



Password Attack Example

- L0phtCrack (“loft-crack”) takes the hashes of passwords and generates the plaintext passwords from them.
- Passwords are compromised using one of two methods:
 - Dictionary cracking
 - Brute-force computation



Trust Exploitation

- Trust exploitation refers to an individual taking advantage of a trust relationship within a network.
- An example of when trust exploitation takes place is when a perimeter network is connected to a corporate network.
 - These network segments often contain DNS, SMTP, and HTTP servers.
 - Because these servers all reside on the same segment, a compromise of one system can lead to the compromise of other systems if those other systems also trust systems that are attached to the same network.

Port Redirection

- A port redirection attack is a type of trust exploitation attack that uses a compromised host to pass traffic through a firewall that would otherwise have been dropped.
 - Port redirection bypasses the firewall rule sets by changing the normal source port for a type of network traffic.
 - You can mitigate port redirection by using proper trust models that are network-specific.
 - Assuming a system is under attack, an IPS can help detect a hacker and prevent installation of such utilities on a host.

“Man-in-the-Middle” Attacks

- Man-in-the-middle attacks have these purposes:
 - Theft of information
 - Hijacking of an ongoing session to gain access to your internal network resources
 - Traffic analysis to obtain information about your network and network users
 - DoS
 - Corruption of transmitted data
 - Introduction of new information into network sessions
- An example of a man-in-the-middle attack is when someone working for your ISP gains access to all network packets that transfer between your network and any other network.

Types of Attacks

- There are four categories of attacks:
 - Malicious Code: Viruses, Worms and Trojan Horses
 - Reconnaissance Attacks
 - Access Attacks
 - Denial of Service (DoS) Attacks

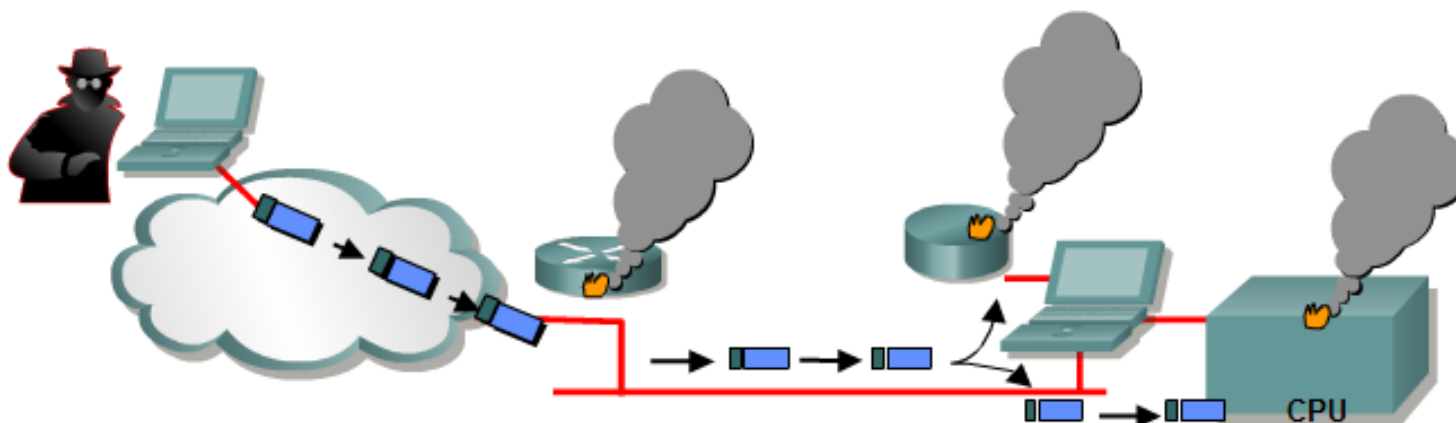
Let's focus on Malicious Code

Denial of Service Attack (DoS)

- Among the most difficult to completely eliminate because they require so little effort to execute.
- Types of DoS attacks include:
 - Ping of death
 - Smurf Attack
 - TCP SYN flood attack
- Others include packet fragmentation and reassembly, E-mail bombs, CPU hogging, Malicious applets, Misconfiguring routers, the chargen attack, out-of-band attacks such as WinNuke, Land.c, Teardrop.c, and Targa.c.

DoS Attacks

DoS attacks prevent authorized people from using a service by using up system resources.



Resource overloads

- Disk space, bandwidth, buffers, and so on.
- Ping floods: smurf, and so on.
- Packet storms: UDP bombs, fraggle, and so on.

Malformed data

- Oversized packets: ping of death, and so on.
- Overlapping packets: winuke, and so on.
- Un-handled data: teardrop, and so on.

Ping of death

- Legacy attack that sent an echo request in an IP packet larger than the maximum packet size of 65,535 bytes.
 - Sending a ping of this size can crash the target computer.
- A variant of this attack is to crash a system by sending ICMP fragments, which fill the reassembly buffers of the target.

Smurf Attack

- This attack sends a large number of ICMP requests to directed broadcast addresses, all with spoofed source addresses on the same network as the respective directed broadcast.
 - If the routing device delivering traffic to those broadcast addresses forwards the directed broadcasts, all hosts on the destination networks send ICMP replies, multiplying the traffic by the number of hosts on the networks.
 - On a multi-access broadcast network, hundreds of machines might reply to each packet.

SYN Flood Attack

- A flood of TCP SYN packets is sent, often with a forged sender address.
 - Each packet is handled like a connection request, causing the server to spawn a half-open (embryonic) connection by sending back a TCP SYN-ACK packet and waiting for a packet in response from the sender address.
 - However, because the sender address is forged, the response never comes.
 - These half-open connections saturate the number of available connections the server is able to make, keeping it from responding to legitimate requests until after the attack ends.

DoS and DDoS Attacks and Mitigation

- A DDoS attack and the simpler version of a DoS attack on a server, send extremely large numbers of requests over a network or the Internet.
 - These many requests cause the target server to run well below optimum speeds.
 - Consequently, the attacked server becomes unavailable for legitimate access and use.
 - By overloading system resources, DoS and DDoS attacks crash applications and processes by executing exploits or a combination of exploits.
 - DoS and DDoS attacks are the most publicized form of attack and are among the most difficult to completely eliminate.

DDoS Attack Example

1. Scan for systems to hack.

Client System

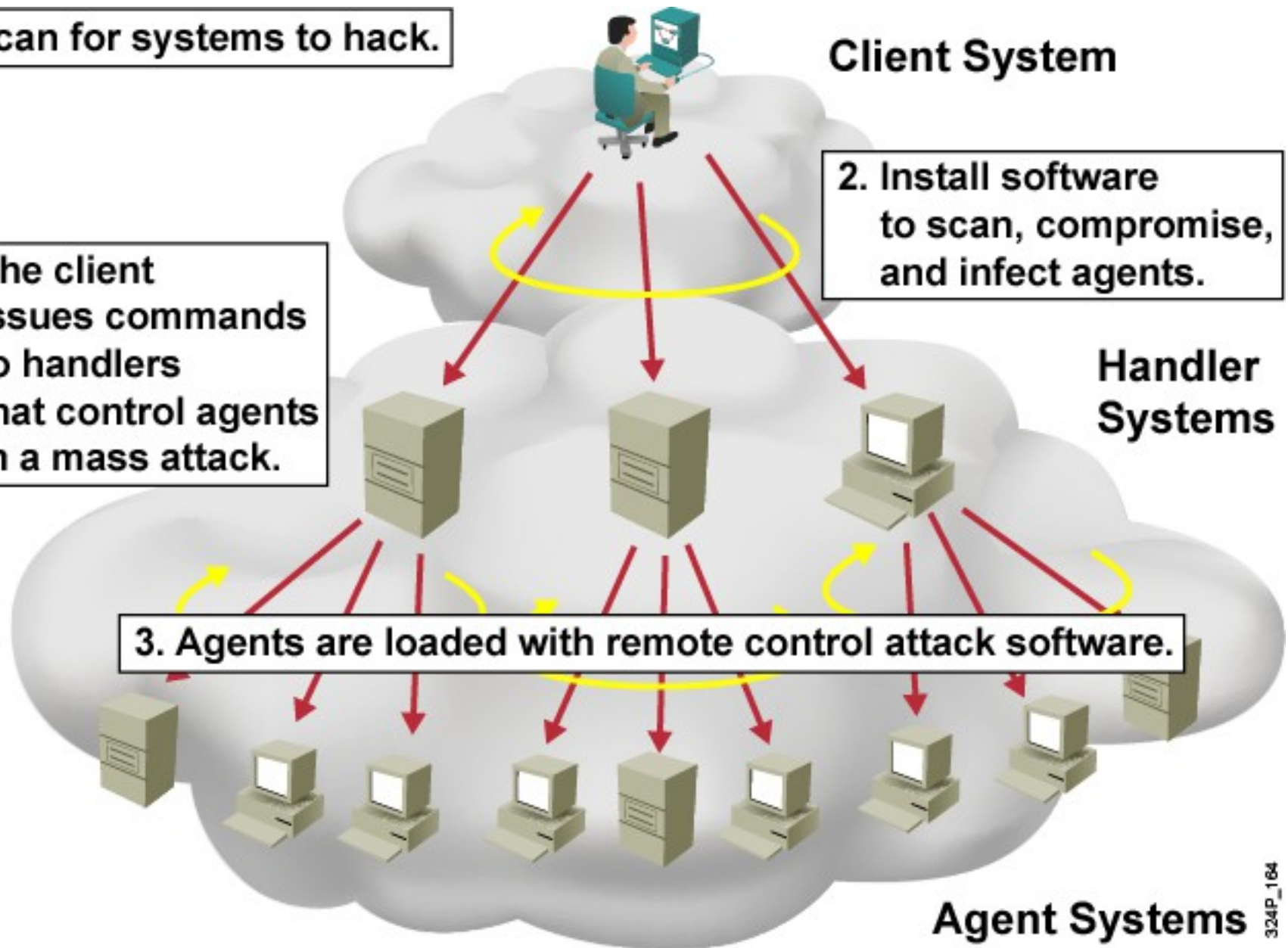
2. Install software to scan, compromise, and infect agents.

Handler Systems

4. The client issues commands to handlers that control agents in a mass attack.

3. Agents are loaded with remote control attack software.

Agent Systems



DDoS Attack Risks

- DDoS attack risks include:
 - Downtime and productivity loss
 - Revenue loss from sales and support services
 - Lost customer loyalty
 - Theft of information
 - Extortion (şantaj)
 - Stock price manipulation
 - Malicious competition

Distributed Denial of Service Attack (DoS)

- DDoS attacks are designed to saturate network links with spurious data which can overwhelm a link causing legitimate traffic to be dropped.
 - DDoS uses attack methods similar to standard DoS attacks but operates on a much larger scale.
 - Typically hundreds or thousands of attack points attempt to overwhelm a target.
- Examples of DDoS attacks include the following:
 - Tribe Flood Network (TFN)
 - Stacheldraht

Reconnaissance Attacks - Countermeasures

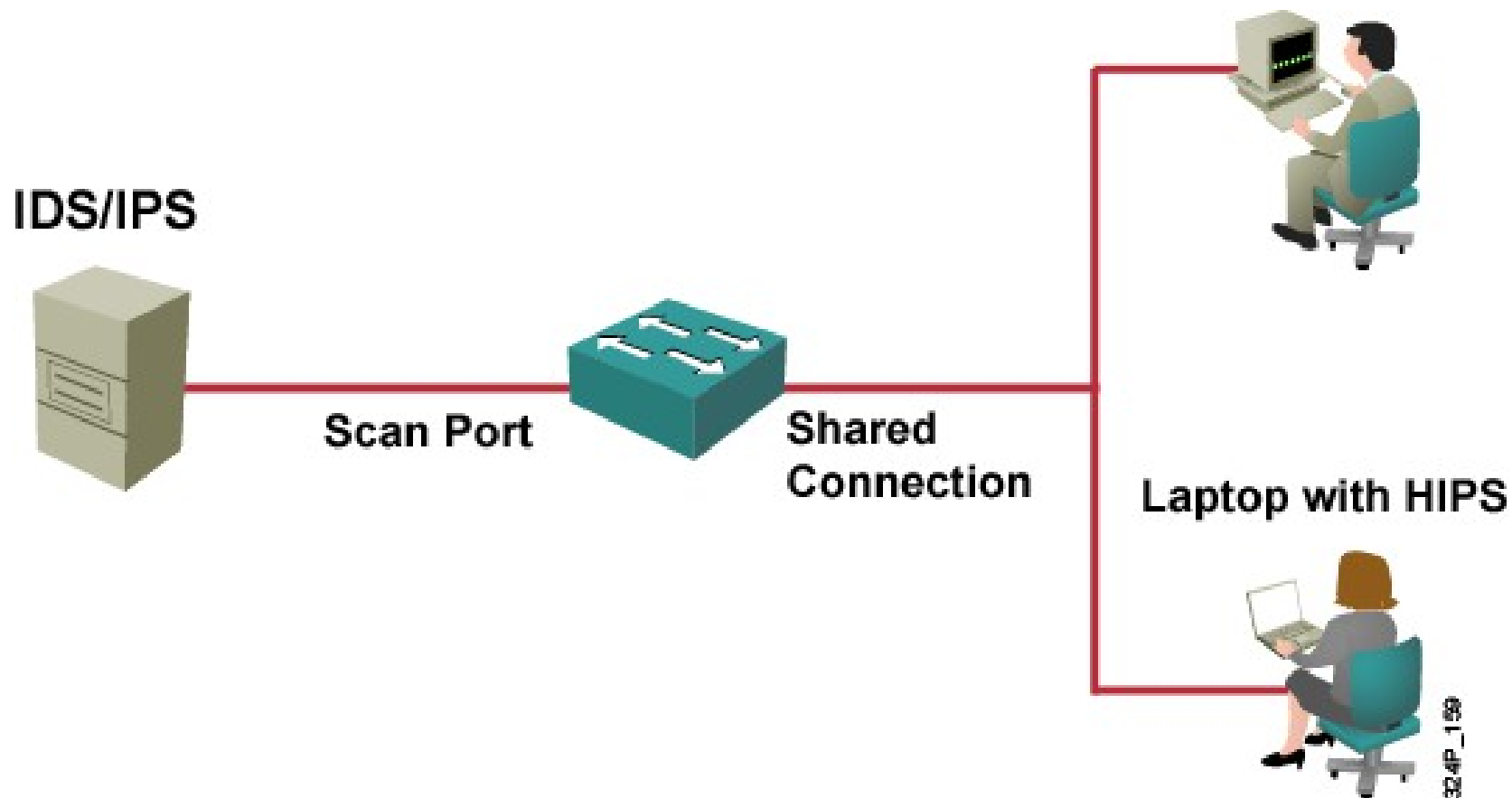
- Implementing and enforcing a policy directive that forbids the use of protocols with known susceptibilities to eavesdropping.
- Using encryption that meets the data security needs of the organization without imposing an excessive burden on the system resources or the users.
- Using switched networks.

Port Scan and Ping Sweep Mitigation (Azaltma)

- Port scanning and ping sweeping is not a crime and there is no way to stop these scans and sweeps when a computer is connected to the Internet.
 - There are ways to prevent damage to the system.
- Ping sweeps can be stopped if ICMP echo and echo-reply are turned off on edge routers.
 - When these services are turned off, network diagnostic data is lost.

Ping Sweeps and Port Scans Mitigation

- Can't be prevented without compromising network capabilities.
 - However, damage can be mitigated using intrusion prevention systems (IPS) at network and host levels.



Packet Sniffer Mitigation

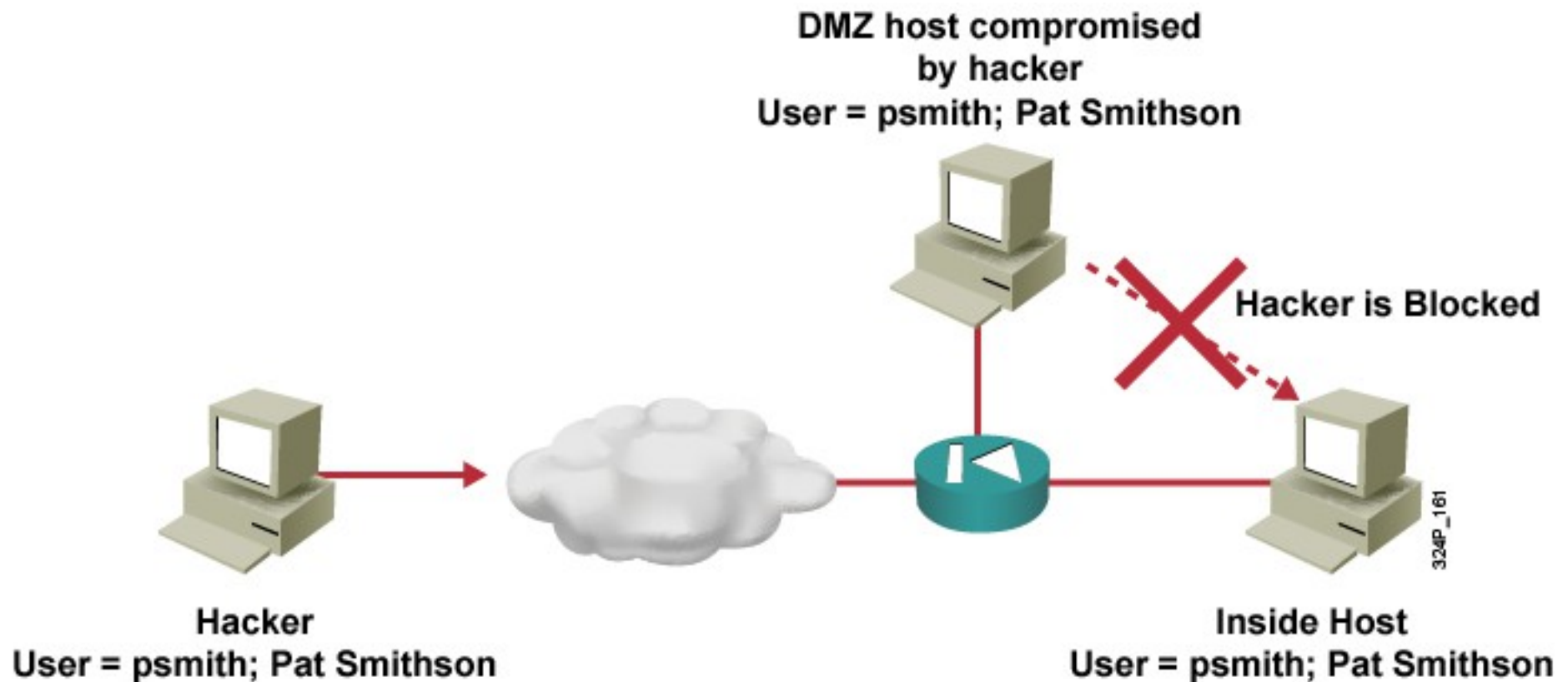
- Authentication
 - Strong authentication is a first line for defense.
- Cryptography
 - If a communication channel is cryptographically secure, the only data a packet sniffer detects is cipher text.
- Anti-sniffer tools
 - Antisniffer tools detect changes in the response time of hosts to determine whether the hosts are processing more traffic than their own traffic loads would indicate.
- Switched infrastructure
 - A switched infrastructure obviously does not eliminate the threat of packet sniffers but can greatly reduce the sniffers' effectiveness.

Password Attack Mitigation

- Password attack mitigation techniques include:
 - Do not allow users to use the same password on multiple systems.
 - Disable accounts after a certain number of unsuccessful login attempts.
 - Use OTP or a cryptographic password is recommended.
 - Use “strong” passwords that are at least eight characters long and contain uppercase letters, lowercase letters, numbers, and special characters.
 - Do not use plain text passwords.

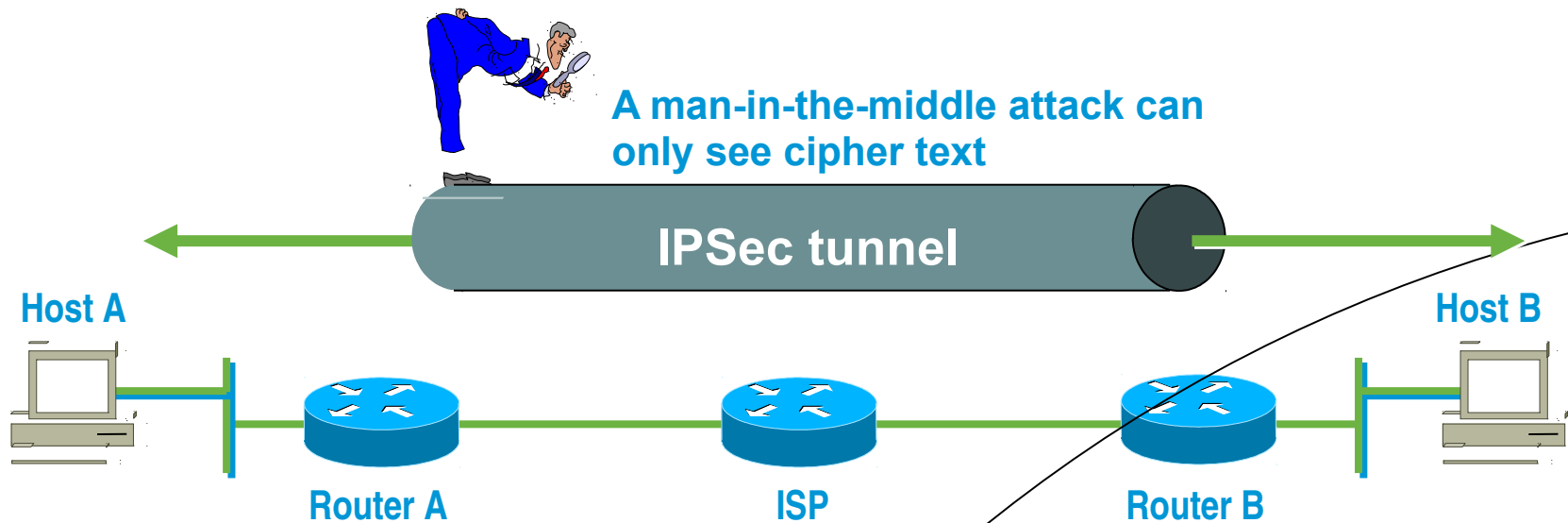
Trust Exploitation Attack Mitigation

- Trust levels within a network should be tightly restrained by ensuring that systems inside a firewall never absolutely trust systems outside the firewall.



Man-in-the-Middle Mitigation

- Man-in-the-middle attacks can be effectively mitigated only through the use of cryptography (encryption).



DoS and DDoS Attack Mitigation

- Anti-DoS features on routers and firewalls:
 - Proper configuration of anti-DoS features on routers and firewalls can help limit the effectiveness of an attack.
 - These features often involve limits on the amount of half-open TCP connections that a system allows at any given time.
- Anti-spoof features on routers and firewalls:
 - Proper configuration of anti-spoof features on your routers and firewalls can reduce your risk of attack.
 - These features include an appropriate filtering with access lists, unicast reverse path forwarding that looks up the routing table to identify spoofed packets, disabling of source route options, and others.

DoS and DDoS Attack Mitigation

- Anti-DoS features on routers and firewalls:
 - Proper configuration of anti-DoS features on routers and firewalls can help limit the effectiveness of an attack.
 - These features often involve limits on the amount of half-open TCP connections that a system allows at any given time.
- Anti-spoof features on routers and firewalls:
 - Proper configuration of anti-spoof features on your routers and firewalls can reduce your risk of attack.
 - These features include an appropriate filtering with access lists, unicast reverse path forwarding that looks up the routing table to identify spoofed packets, disabling of source route options, and others.

IP Spoofing Attack Mitigation

- The threat of IP spoofing can be reduced, but not eliminated, using these measures:
 - Access control configuration
 - Encryption
 - RFC 3704 filtering
- Additional authentication requirement that does not use IP address-based authentication; examples are:
 - Cryptographic (recommended)
 - Strong, two-factor, one-time passwords

10 Best Practices

1. Keep patches up to date by installing them weekly or daily, if possible, to prevent buffer overflow and privilege escalation attacks.
2. Shut down unnecessary services and ports.
3. Use strong passwords and change them often.
4. Control physical access to systems.
5. Avoid unnecessary web page inputs.
 - Some websites allow users to enter usernames and passwords.
 - A hacker can enter more than just a username.
 - For example, entering "jdoe; rm -rf /" might allow an attacker to remove the root file system from a UNIX server.
 - Programmers should limit input characters and not accept invalid characters such as | ; < > as input.

10 Best Practices

1. Keep patches up to date by installing them weekly or daily, if possible, to prevent buffer overflow and privilege escalation attacks.
2. Shut down unnecessary services and ports.
3. Use strong passwords and change them often.
4. Control physical access to systems.
5. Avoid unnecessary web page inputs.
 - Some websites allow users to enter usernames and passwords.
 - A hacker can enter more than just a username.
 - For example, entering "jdoe; rm -rf /" might allow an attacker to remove the root file system from a UNIX server.
 - Programmers should limit input characters and not accept invalid characters such as | ; < > as input.

Know Thine Enemy



- "If you know yourself but not your enemy, for every victory gained you will also suffer a defeat."
 - Sun Tzu – The Art of War
- Before learning how to defend against attacks, you need to know how a potential attacker operates.

Hacking a Network

- The goal of any hacker is to compromise the intended target or application.
- Hackers begin with little or no information about the intended target.
- Their approach is always careful and methodical—never rushed and never reckless.
- The seven-step process outlined on the next slide is a good representation of the method that hackers use – and a starting point for an analysis of how to defeat it.

Seven Steps to Hacking a Network

- Step 1 — Perform footprint analysis (reconnaissance).
- Step 2 — Detail the information.
- Step 3 — Manipulate users to gain access.
- Step 4 — Escalate privileges.
- Step 5 — Gather additional passwords and secrets.
- Step 6 — Install back doors.
- Step 7 — Leverage the compromised system.

Step 1 - Footprint Analysis (Reconnaissance)

- Gain knowledge of acquisitions using Web pages, phone books, company brochures, subsidiaries, etc.
- Use commands to develop a more detailed footprint:
 - **nslookup** command to reconcile domain names against IP addresses of the company's servers and devices.
 - **tracert** command to help build topology.
- Use program and utilities:
 - **WHOIS** queries (<http://www.who.is/>)
 - Port scanning to find open ports and operating systems installed on hosts.
 - **Nmap**: Network Mapper (Nmap) is a free open source utility for network exploration or security auditing.

How to Defeat Footprinting

- Keep all sensitive data off-line (business plans, formulas, and proprietary documents).
- Minimize the amount of information on your public website.
- Examine your own website for insecurities.
- Run a ping sweep on your network.
- Familiarize yourself with one or more of the five Regional Internet Registries – such as ARIN for North America – to determine network blocks.

Step 2 - Detail the Information

- Find your server applications and versions:
 - What are your web, FTP, and mail server versions?
 - Listen to TCP and UDP ports and send random data to each.
 - Cross-reference information to vulnerability databases to look for potential exploits.
- Exploit selected TCP ports, for example:
 - Windows NT, 2000, and XP file sharing using SMB protocol which uses TCP port 445.
 - In Windows NT, SMB runs on top of NetBT using ports 137, 138 (UDP), and 139 (TCP).

Software Tools

- A great deal of hacker tools are available:
 - Netcat: Netcat is a featured networking utility that reads and writes data across network connections using the TCP/IP protocol.
 - Microsoft EPDump and Remote Procedure Call (RPC) Dump: These tools provide information about Microsoft RPC services on a server:
 - The Microsoft EPDump application shows what is running and waiting on dynamically assigned ports.
 - The RPC Dump (rpcdump.exe) application is a command-line tool that queries RPC endpoints for status and other information on RPC.
 - GetMAC: This application provides a quick way to find the MAC (Ethernet) layer address and binding order for a computer running Microsoft Windows 2000 locally or across a network.
 - Software development kits (SDKs): SDKs provide hackers with the basic tools that they need to learn more about systems.

Step 3 - Manipulate Users to Gain Access

- Even with the most sophisticated security in place, a company is still vulnerable because of security's weakest link: People!
- The first thing that hackers need is a password and there are two ways to get that password:
 - Social engineering
 - Password cracking attacks

Social Engineering Example #1

- Call in the middle of the night:
 - ‘Hi this is _____ from Bell. I’m very sorry to wake you up but we’ve noticed some very unusual activity on your Bell calling card and we’re wondering if you’re using it to call Baghdad, Iraq for the last 6 hours?’
 - ‘Well, we have a call that’s actually still active right now and it’s now well over \$2,000 worth of charges. I’ll terminate that call right now but unfortunately you are responsible for the charges made on your card.’
 - ‘Look I sympathize with you and can see that you’ve been victimized here, but if I get rid of that charge I can loose my job.’
 - ‘Okay ... but you’ll have to confirm some details first. What is your full name and address?’
 - ‘Can you confirm the Bell calling card number?’
 - ‘Finally, please confirm your PIN number?’
 - ‘Great. Everything matches. I’ll get rid of that charge for you.’
 - ‘You’re welcome and thank you for being a Bell Canada client.’

Social Engineering Example #2

- The facilitator of a live Computer Security Institute neatly illustrated the vulnerability of help desks when he “dialed up” a phone company, got transferred around, and reached the help desk:
 - ‘Who’s the supervisor on duty tonight?’
 - ‘Let me talk to _____.’ (he’s transferred)
 - ‘Hi _____, this is _____ from security in the IT center. Having a bad day?’
 - ‘No, why?...Your systems are down.’
 - Response: ‘my systems aren’t down, we’re running fine.’
 - ‘Hmmm ... Really? Do me a favor then and sign off and on again.’
 - ‘We didn’t even show a blip, we show no change. Sign off again.’
 - ‘There’s something funny going on here. I’m going to have to sign on with your ID to figure out what’s happening. Let me have your user ID and password.’

Other Social Engineering Examples

- A confused and befuddled person will call a clerk and meekly request a password change.
- People identifying themselves as executives, will telephone a new system administrator and demand access to their account IMMEDIATELY!
- Somebody will call and confidently instruct a computer operator to type in a few lines of instruction at the console.
- At an airport, somebody will look over a shoulder, 'shoulder surfing,' (sometimes even using binoculars or camcorders) as telephone credit card numbers or ATM PINs are keyed.

Common Social Engineering Methods

- Using insider lingo and terminology to gain trust.
- Offering a prize for registering at a Web site with username and password.
- Dropping a document or file at company mail room for intra-office delivery.
- Modifying fax machine heading to appear to come from an internal location.
- Asking receptionist to receive then forward a fax.
- Asking for a file to be transferred to an apparently internal location.
- Getting a voice mailbox set up so call backs perceive attacker as internal.
- Pretending to be from remote office and asking for email access locally.

Common Social Engineering Methods

- Using insider lingo and terminology to gain trust.
- Offering a prize for registering at a Web site with username and password.
- Dropping a document or file at company mail room for intra-office delivery.
- Modifying fax machine heading to appear to come from an internal location.
- Asking receptionist to receive then forward a fax.
- Asking for a file to be transferred to an apparently internal location.
- Getting a voice mailbox set up so call backs perceive attacker as internal.
- Pretending to be from remote office and asking for email access locally.

Warning Signs of an Attack

- Refusal to give call back number
- Out-of-ordinary request
- Claim of authority
- Stresses urgency
- Threatens negative consequences of non compliance
- Shows discomfort when questioned
- Name dropping
- Compliments or flattery
- Flirting

Password Cracking

- Hackers use many tools and techniques to crack passwords:
 - Word lists
 - Brute force
 - Hybrids
 - The yellow Post-It stuck on the side of the monitor, or in top of desk drawer
- Password cracking attacks any application or service that accepts user authentication, including those listed here:
 - NetBIOS over TCP (TCP 139)
 - Direct host (TCP 445)
 - FTP (TCP 21)
 - Telnet (TCP 23)
 - SNMP (UDP 161)
 - PPTP (TCP 1723)
 - Terminal services (TCP 3389)

Step 4 - Escalate Privileges

- After securing a password for a user account and user-level privileges to a host, hackers attempt to escalate their privileges.
- The hacker will review all the information he or she can see on the host:
 - Files containing user names and passwords
 - Registry keys containing application or user passwords
 - Any available documentation (for example, e-mail)
- If the host cannot be seen by the hacker, the hacker may launch a Trojan application such as W32/QAZ to provide it.

Step 5 – Gather Passwords and Secrets

- Hackers target:
 - The local security accounts manager database
 - The active directory of a domain controller
- Hackers can use legitimate tools including pwdump and lsadump applications.
- Hackers gain administrative access to all computers by cross-referencing user names and password combinations.

Step 6 - Install Back Doors and Port Redirectors

- Back doors:
 - Provide a way back into the system if the front door is locked.
 - The way into the system that is not likely to be detected.
- Back doors may use reverse trafficking:
 - Example: Code Red which used TCP port 80 to instruct unpatched web servers to execute a TFTP connection from the server.
- Port redirectors:
 - Port redirectors can help bypass port filters, routers, and firewalls and may even be encrypted over an SSL tunnel to evade intrusion detection devices.



Step 7 - Leverage the Compromised System

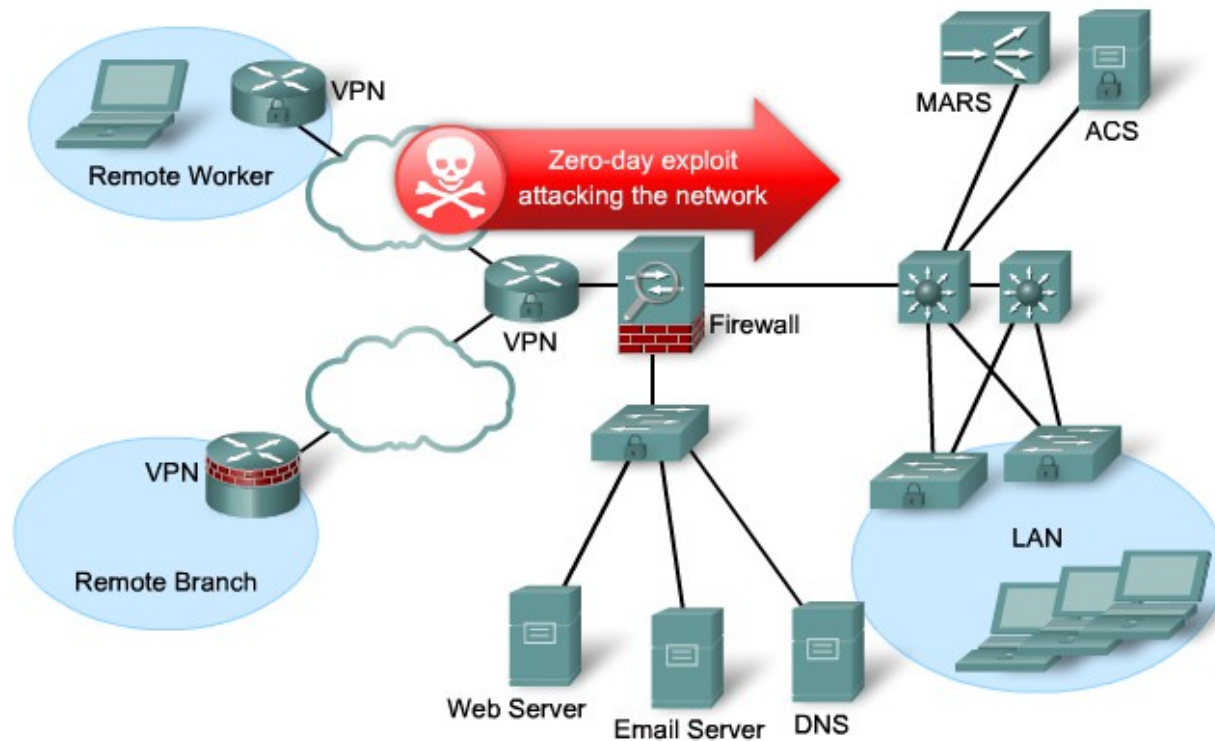
- Back doors and port redirectors let hackers attack other systems in the network.
- Reverse trafficking lets hackers bypass security mechanisms.
- Trojans let hackers execute commands undetected.
- Scanning and exploiting the network can be automated.
- The hacker remains behind the cover of a valid administrator account.
- The whole seven-step process is repeated as the hacker continues to penetrate the network.

Best Practices to Defeat Hackers

- Keep patches up to date.
- Shut down unnecessary services and ports.
- Use strong passwords and change them often.
- Control physical access to systems.
- Avoid unnecessary web page inputs.
 - Some websites allow users to enter usernames and passwords.
 - A hacker can enter more than just a username and programmers should limit input characters and not accept invalid characters (| ; < >).
- Perform system backups and test them on a regular basis.
- Educate users about social engineering.
- Encrypt and password-protect sensitive data.
- Use appropriate security hardware and software.
- Develop a written security policy for the company.

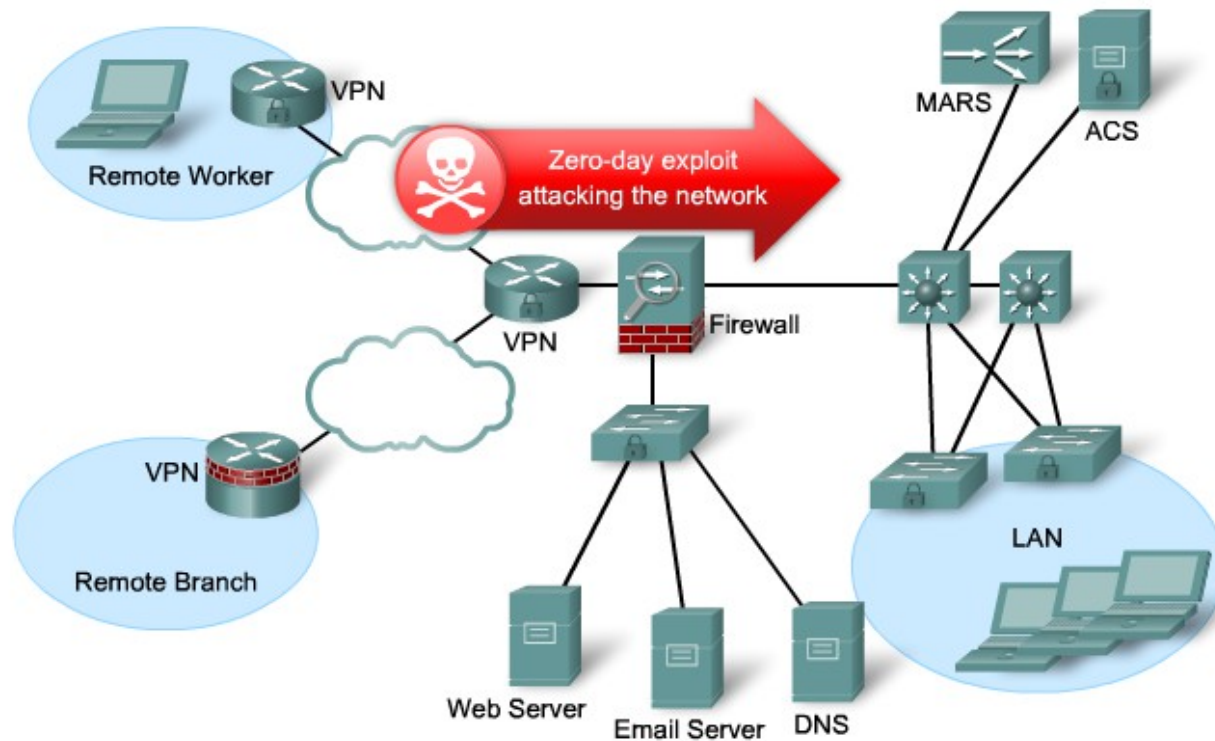
Zero-Day Exploits

- Worms and viruses can spread across the world in minutes.
 - **Zero-day attack** (zero-day threat), is a computer attack that tries to exploit software vulnerabilities.
 - **Zero-hour** describes the moment when the exploit is discovered.



Zero-Day Exploits

- Worms and viruses can spread across the world in minutes.
 - **Zero-day attack** (zero-day threat), is a computer attack that tries to exploit software vulnerabilities.
 - **Zero-hour** describes the moment when the exploit is discovered.



How do you protect your computer?

- Do you constantly:
 - Sit there looking at Task Manager for nefarious processes?
 - Look at the Event Viewer logs looking for anything suspicious?
- You rely on anti-virus software and firewall features.

Intrusion Prevention

The ability to stop attacks against the network and provide the following active defense mechanisms:

Detection – Identifies malicious attacks on network and host resources.

Prevention – Stops the detected attack from executing.

Reaction – Immunizes the system from future attacks from a malicious source.

Either technology can be implemented at a network level, host level, or both for maximum protection.

Examples of Intrusion

- Performing a remote root compromise of an e-mail server
- Defacing a Web server
- Guessing and cracking passwords
- Copying a database containing credit card numbers
- Viewing sensitive data, including payroll records and medical information, without authorization
- Running a packet sniffer on a workstation to capture usernames and passwords
- Using a permission error on an anonymous FTP server to distribute pirated software and music files
- Dialing into an unsecured modem and gaining internal network access
- Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password
- Using an unattended, logged-in workstation without permission

Hackers

- Traditionally, those who hack into computers do so for the thrill of it or for status
- Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter hacker threats
 - In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology
- CERTs
 - Computer emergency response teams
 - These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers
 - Hackers also routinely read CERT reports
 - It is important for system administrators to quickly insert all software patches to discovered vulnerabilities

Criminal hackers

- Organized groups of hackers
- Usually have specific targets, or at least classes of targets in mind
- Once a site is penetrated, the attacker acts quickly, scooping up as much valuable information as possible and exiting
- IDSs and IPSs can be used for these types of attackers, but may be less effective because of the quick in-and-out nature of the attack

- Among the most difficult to detect and prevent
- Can be motivated by revenge or simply a feeling of entitlement
- Countermeasures:
 - Objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system
 - Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a backdoor into the system
 - Ways to protect a password file:

Password Guessing

1. Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change these defaults.
2. Exhaustively try all short passwords (those of one to three characters).
3. Try words in the system's online dictionary or a list of likely passwords. Examples of the latter are readily available on hacker bulletin boards.
4. Collect information about users, such as their full names, the names of their spouse and children, pictures in their office, and books in their office that are related to hobbies.
5. Try users' phone numbers, Social Security numbers, and room numbers.
6. Try all legitimate license plate numbers for this state.
7. Use a Trojan horse to bypass restrictions on access.
8. Tap the line between a remote user and the host system.

Intrusion Detection

- A system's second line of defense
- Is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified
- Considerations:
 - If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised
 - An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions
 - Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility

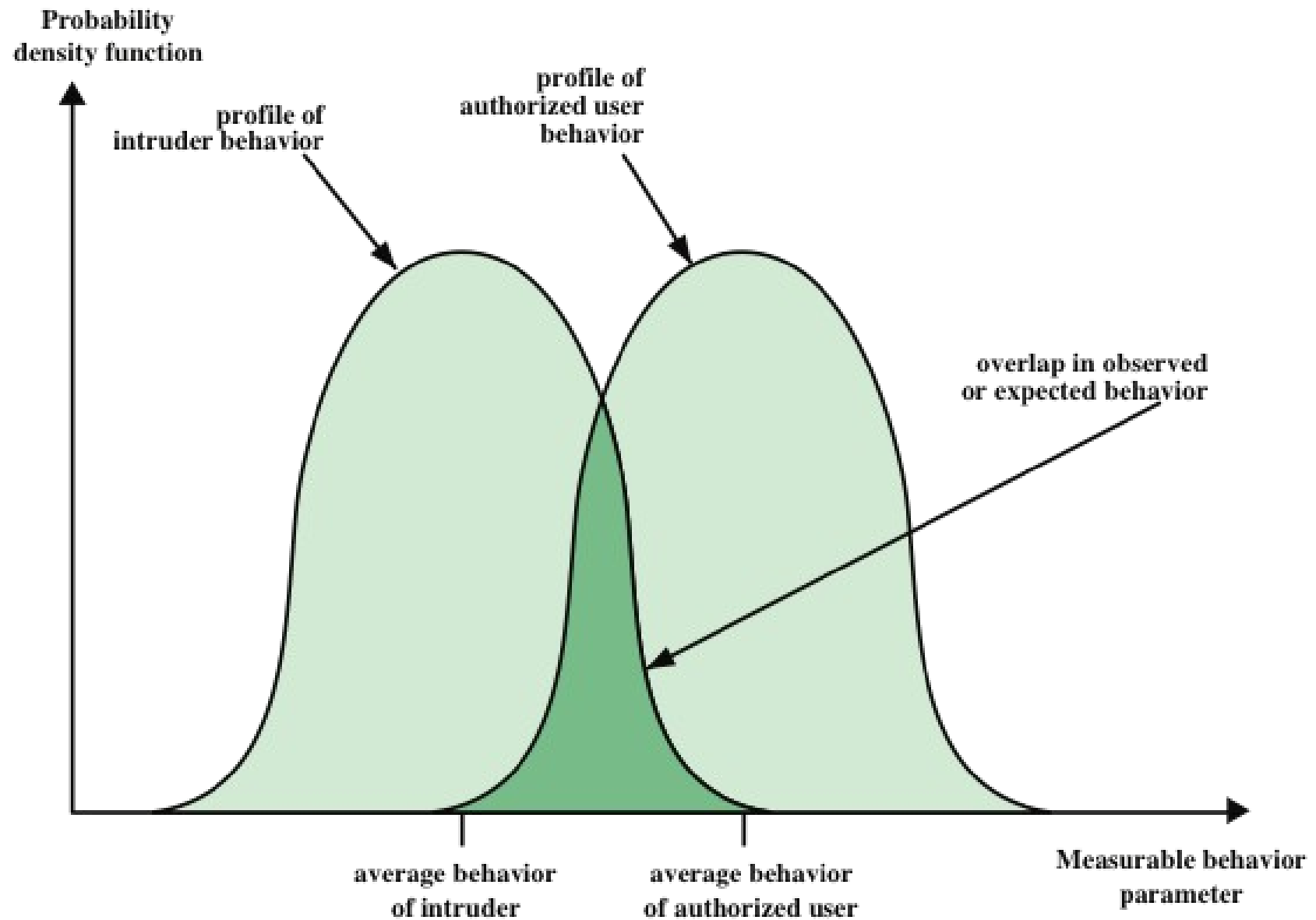


Figure 11.1 Profiles of Behavior of Intruders and Authorized Users

Approaches to Intrusion Detection

- Statistical anomaly detection
 - Involves the collection of data relating to the behavior of legitimate users over a period of time
 - Then statistical tests are applied to observed behavior to determine whether that behavior is not legitimate user behavior
 - Threshold detection
 - This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events
 - Profile based
 - A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts
- Rule-based detection
 - Involves an attempt to define a set of rules or attack patterns that can be used to decide that a given behavior is that of an intruder
 - Often referred to as *signature detection*

Statistical Anomaly Detection

- Threshold detection
 - Involves counting the number of occurrences of a specific event type over an interval of time
 - If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed
 - By itself is a crude and ineffective detector of even moderately sophisticated attacks
- Profile-based
 - Focuses on characterizing the past behavior of individual users or related groups of users and then detecting significant deviations
 - A profile may consist of a set of parameters, so that deviation on just a single parameter may not be sufficient in itself to signal an alert

Measure	Model	Type of Intrusion Detected
Login and Session Activity		
Login frequency by day and time	Mean and standard deviation	Intruders may be likely to log in during off-hours.
Frequency of login at different locations	Mean and standard deviation	Intruders may log in from a location that a particular user rarely or never uses.
Time since last login	Operational	Break-in on a "dead" account.
Elapsed time per session	Mean and standard deviation	Significant deviations might indicate masquerader.
Quantity of output to location	Mean and standard deviation	Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data.
Session resource utilization	Mean and standard deviation	Unusual processor or I/O levels could signal an intruder.
Password failures at login	Operational	Attempted break-in by password guessing.
Failures to login from specified terminals	Operational	Attempted break-in.
Command or Program Execution Activity		
Execution frequency	Mean and standard deviation	May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.
Program resource utilization	Mean and standard deviation	An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization.
Execution denials	Operational model	May detect penetration attempt by individual user who seeks higher privileges.
File access activity		
Read, write, create, delete frequency	Mean and standard deviation	Abnormalities for read and write access for individual users may signify masquerading or browsing.
Records read, written	Mean and standard deviation	Abnormality could signify an attempt to obtain sensitive data by inference and aggregation.
Failure count for read, write, create, delete	Operational	May detect users who persistently attempt to access

Rule-Based Intrusion Detection

- Techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious
- **Rule-based anomaly detection**
 - Is similar in terms of its approach and strengths to statistical anomaly detection
 - Historical audit records are analyzed to identify usage patterns and to automatically generate rules that describe those patterns
 - Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior
 - In order for this approach to be effective, a rather large database of rules will be needed

Rule-Based Intrusion Detection

- Techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious
- **Rule-based anomaly detection**
 - Is similar in terms of its approach and strengths to statistical anomaly detection
 - Historical audit records are analyzed to identify usage patterns and to automatically generate rules that describe those patterns
 - Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior
 - In order for this approach to be effective, a rather large database of rules will be needed

USTAT Action	SunOS Event Type
Read	open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt
Write	truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct
Create	mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod
Delete	rmdir, unlink
Execute	exec, execve
Exit	exit
Modify_Owner	chown, fchown
Modify_Perm	chmod, fchmod
Rename	rename
Hardlink	link

Base-Rate Fallacy

- To be of practical use, an intrusion detection system should detect a substantial percentage of intrusions while keeping the false alarm rate at an acceptable level
 - If only a modest percentage of actual intrusions are detected, the system provides a false sense of security
 - If the system frequently triggers an alert when there is no intrusion, then either system managers will begin to ignore the alarms or much time will be wasted analyzing the false alarms
- Because of the nature of the probabilities involved, it is very difficult to meet the standard of high rate of detections with a low rate of false alarms
 - If the actual numbers of intrusions is low compared to the number of legitimate uses of a system, then the false alarm rate will be high unless the test is extremely discriminating
- See Appendix J for a brief background on the mathematics of this problem

Distributed Intrusion Detection

- Traditional systems focused on single-system stand-alone facilities
 - The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN or internetwork
 - A more effective defense can be achieved by coordination and cooperation among intrusion detection systems across the network
- Major design issues:

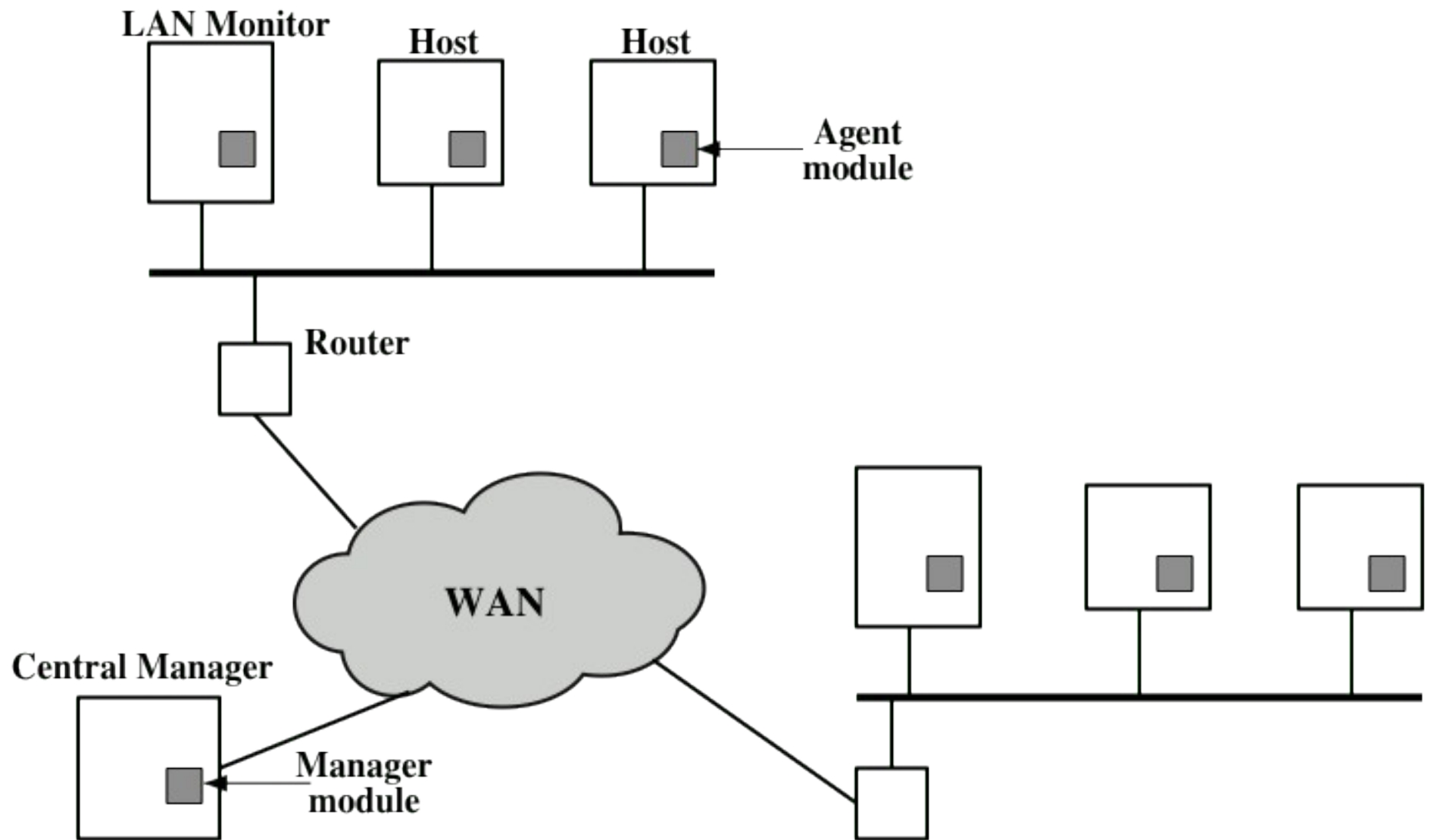


Figure 11.2 Architecture for Distributed Intrusion Detection

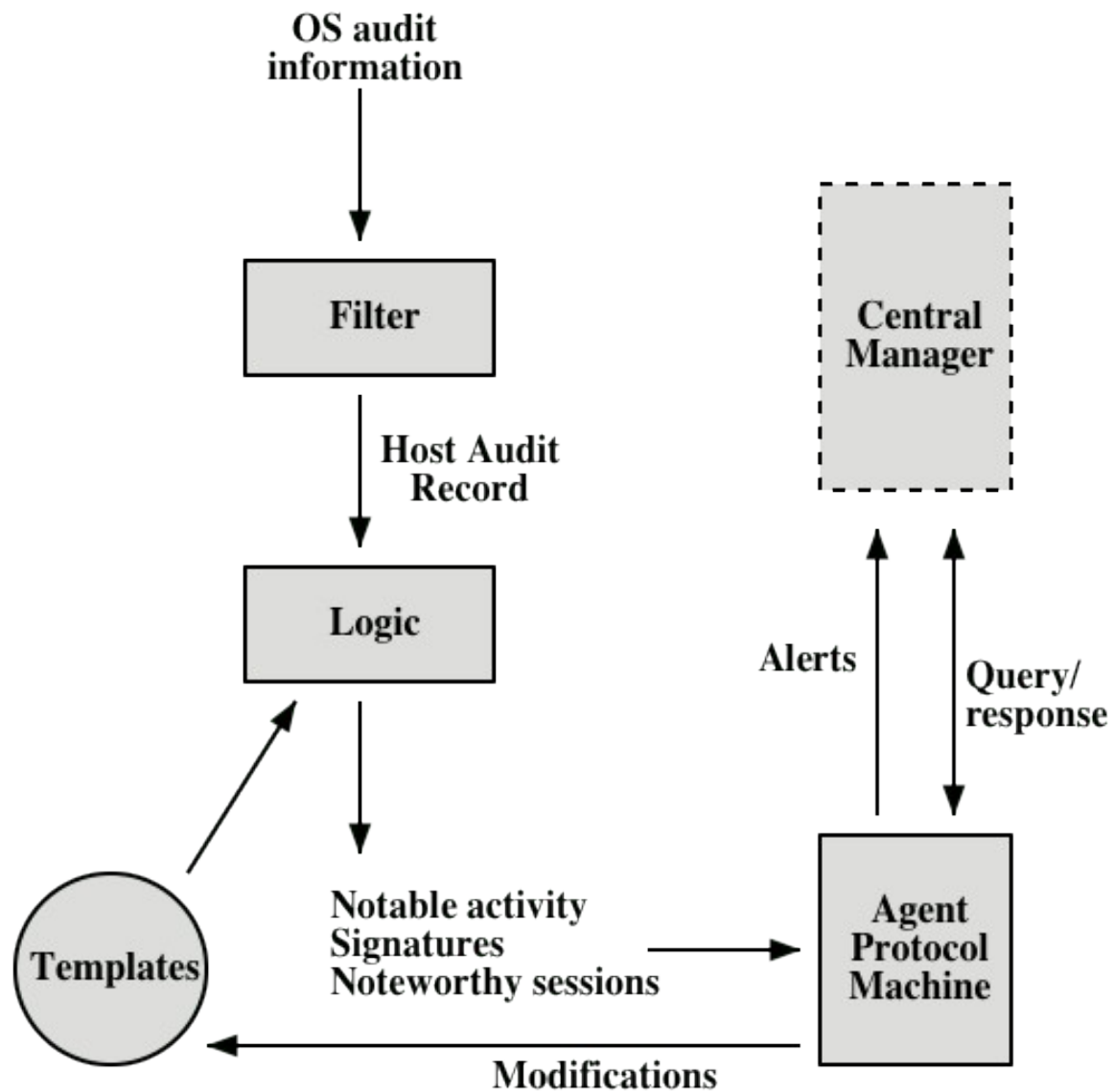


Figure 11.3 Agent Architecture

Honeypots

- Decoy systems that are designed to lure a potential attacker away from critical systems
- Because any attack against the honeypot is made to seem successful, administrators have time to mobilize and log and track the attacker without ever exposing productive systems
- Recent research has focused on building entire honeypot networks that emulate an enterprise, possible with actual or simulated traffic and data

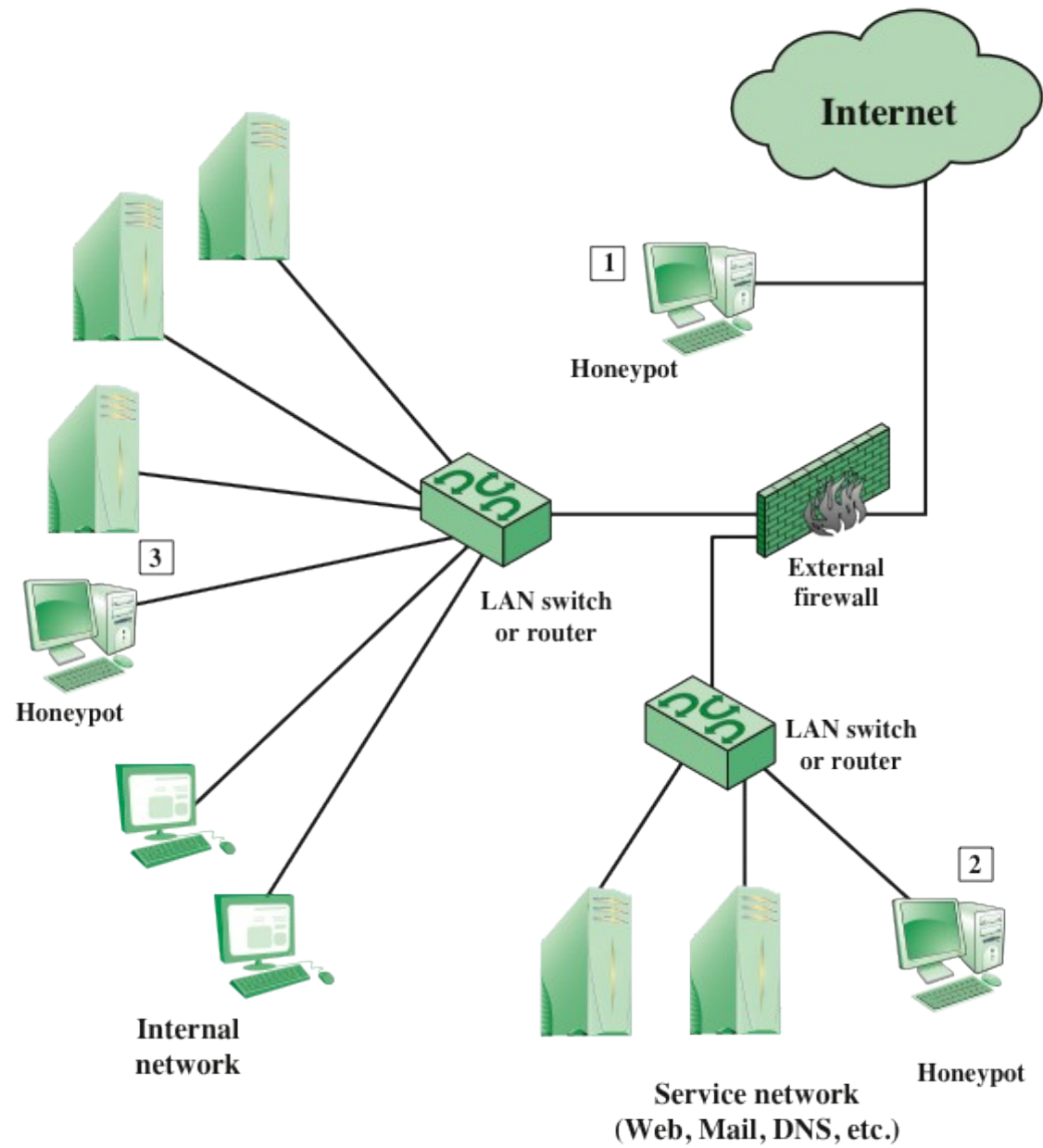


Figure 11.4 Example of Honeypot Deployment

- To facilitate the development of distributed intrusion detection systems that can function across a wide range of platforms and environments, standards are needed to support interoperability
- IETF Intrusion Detection Working Group
 - Purpose of the group is to define data formats and exchange procedures for sharing information of interest to intrusion detection with response systems and to management systems that may need to interact with them
 - Have issued the following RFCs:
 - Intrusion Detection Message Exchange Requirements (RFC 4766)
 - The Intrusion Detection Message Exchange Format (RFC 4765)
 - The Intrusion Detection Exchange Protocol (RFC 4767)

exchange format

- To facilitate the development of distributed intrusion detection systems that can function across a wide range of platforms and environments, standards are needed to support interoperability
- IETF Intrusion Detection Working Group
 - Purpose of the group is to define data formats and exchange procedures for sharing information of interest to intrusion detection with response systems and to management systems that may need to interact with them
 - Have issued the following RFCs:
 - Intrusion Detection Message Exchange Requirements (RFC 4766)
 - The Intrusion Detection Message Exchange Format (RFC 4765)
 - The Intrusion Detection Exchange Protocol (RFC 4767)

- Internet connectivity is no longer optional for organizations
 - Individual users within the organization want and need Internet access
- While Internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets
 - This creates a threat to the organization
 - While it is possible to equip each workstation and server on the premises network with strong security features, this may not be sufficient and in some cases is not cost-effective
- Firewall
 - An alternative, or at least complement, to host-based security services
 - Is inserted between the premises network and the Internet to establish a controlled link and to erect an outer security wall or perimeter
 - The aim of this perimeter is to protect the premises network from Internet-based attacks and to provide a single choke point where security and auditing can be imposed
 - May be a single computer system or a set of two or more systems that cooperate to perform the firewall function

Firewall characteristics

- Design goals for a firewall:
 - All traffic from inside to outside, and vice versa, must pass through the firewall
 - Only authorized traffic, as defined by the local security policy, will be allowed to pass
 - The firewall itself is immune to penetration
- Techniques that firewalls use to control access and enforce the site's security policy:

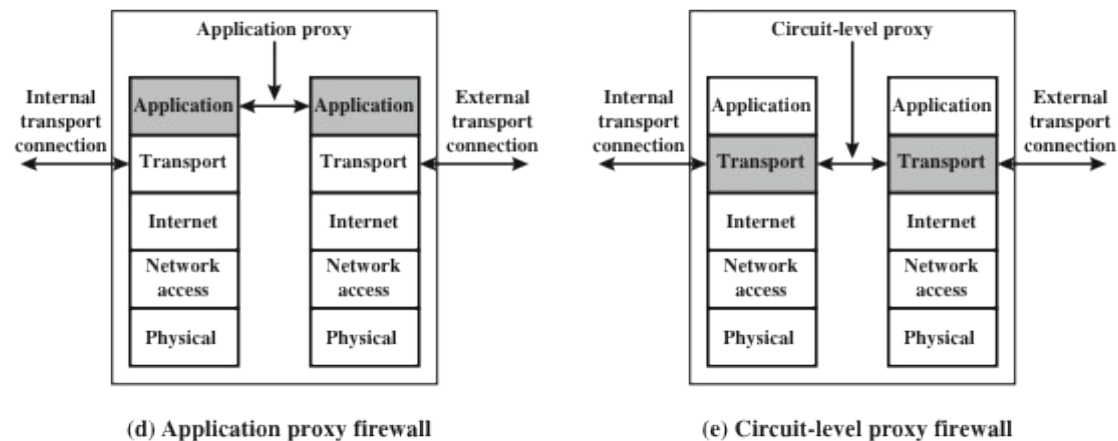
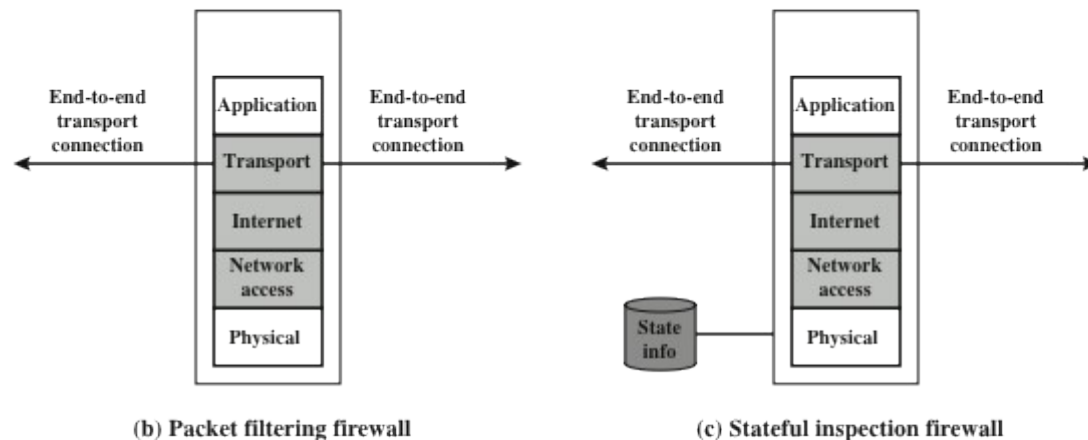
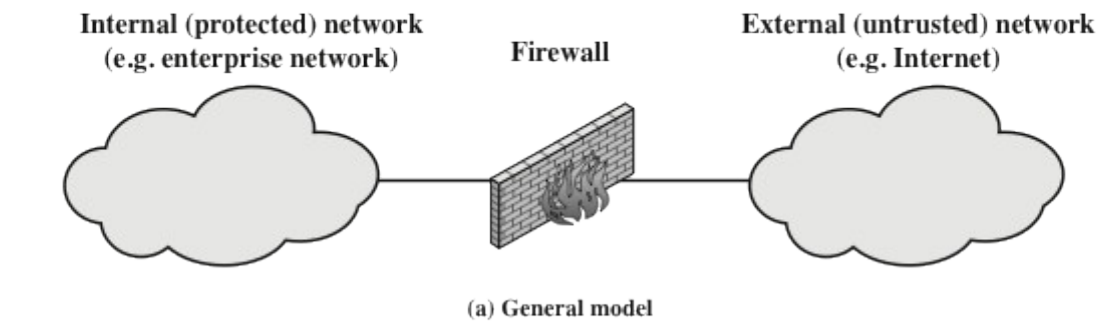


Figure 12.1 Types of Firewalls

Rule	Direction	Src address	Dest addresss	Protocol	Dest port	Action
A	In	External	Internal	TCP	25	Permit
B	Out	Internal	External	TCP	>1023	Permit
C	Out	Internal	External	TCP	25	Permit
D	In	External	Internal	TCP	>1023	Permit
E	Either	Any	Any	Any	Any	Deny

Table 12.1
Packet-Filtering Example

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.22.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	80	Established
2122.22.123.32	2112	192.168.1.6	80	Established
210.922.212.18	3321	192.168.1.6	80	Established
24.102.32.23	1025	192.168.1.6	80	Established
223.21.22.12	1046	192.168.1.6	80	Established

Table 12.2
Example Stateful Firewall Connection State Table
[SCAR09b]

Application Level Gateway

- Also called an *application proxy*
- Acts as a relay of application-level traffic
- If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall
- The gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features
- Tend to be more secure than packet filters
- Disadvantage:
 - The additional processing overhead on each connection

Circuit-Level Gateway

- Also called *circuit-level proxy*
- Can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications
- Does not permit an end-to-end TCP connection
- The security function consists of determining which connections will be allowed
- Typical use is a situation in which the system administrator trusts the internal users
- Can be configured to support application-level or proxy service on inbound connections and circuit-level functions for outbound connections
- Example of implementation is the SOCKS package

Bastion Host

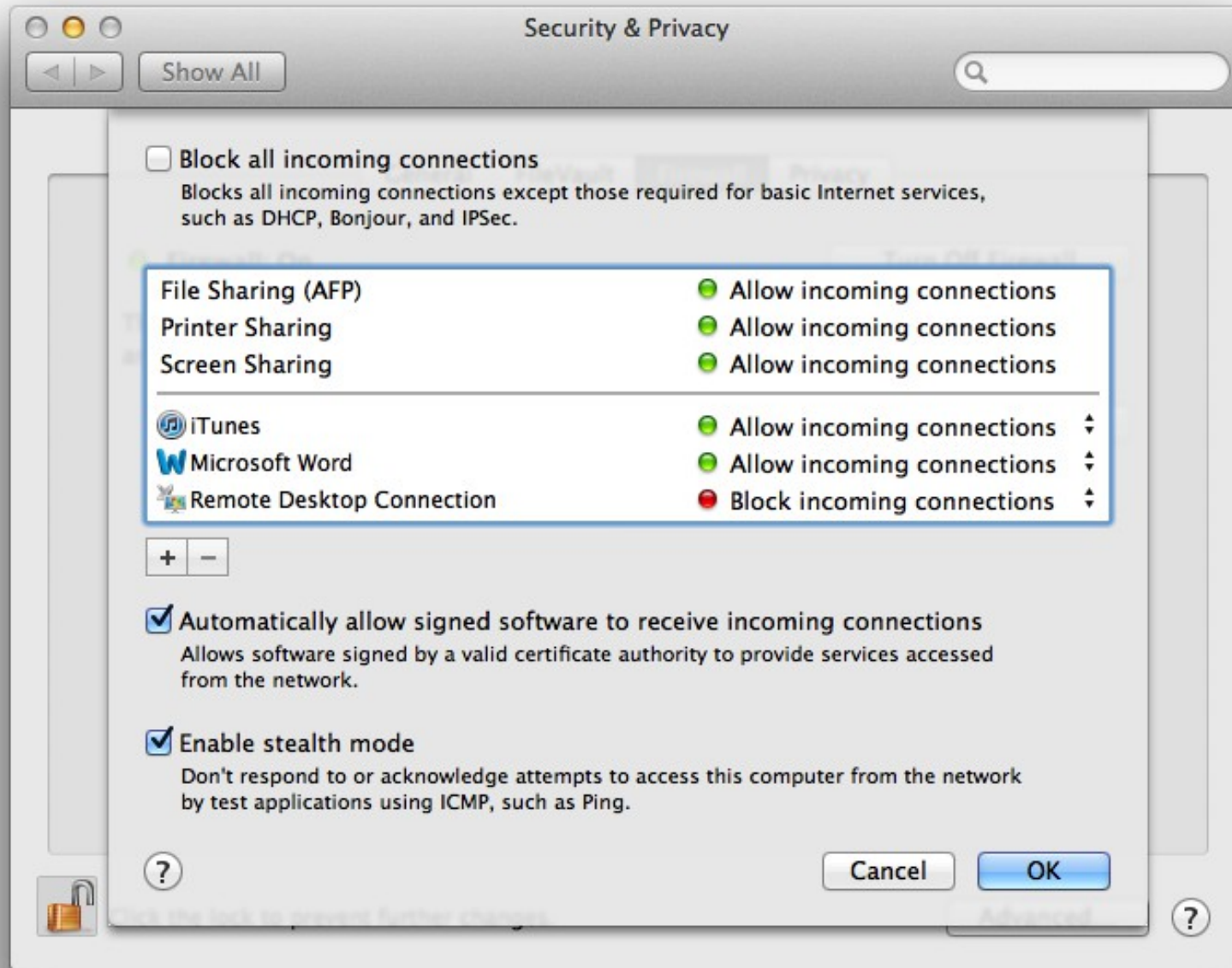
- A system identified by the firewall administrator as a critical strong point in the network's security
- Typically serves as a platform for an application-level or circuit-level gateway
- Common characteristics:
 - Executes a secure version of its operating system, making it a hardened system
 - Only the services that the network administrator considers essential are installed
 - May require additional authentication before a user is allowed access to the proxy services
 - Each proxy is configured to support only a subset of the standard application's command set
 - Each proxy is configured to allow access only to specific host systems
 - Each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection
 - Each proxy module is a very small software package specifically designed for network security
 - Each proxy is independent of other proxies on the bastion host
 - A proxy generally performs no disk access other than to read its initial configuration file
 - Each proxy runs as a nonprivileged user in a private and secured directory on the bastion host

Host-Based Firewall

- A software module used to secure an individual host
- Is available in many operating systems or can be provided as an add-on package
- Filters and restricts the flow of packets
- Common location is a server
- Advantages:
 - Filtering rules can be tailored to the host environment
 - Protection is provided independent of topology
 - Used in conjunction with stand-alone firewalls, provides an additional layer of protection

Personal Firewall

- Controls the traffic between a personal computer or workstation on one side and the Internet or enterprise network on the other side
- Can be used in the home environment and on corporate intranets
- Typically is a software module on the personal computer
- Can also be housed in a router that connects all of the home computers to a DSL, cable modem, or other Internet interface
- Primary role is to deny unauthorized remote access to the computer
- Can also monitor outgoing activity in an attempt to detect and block worms and other malware



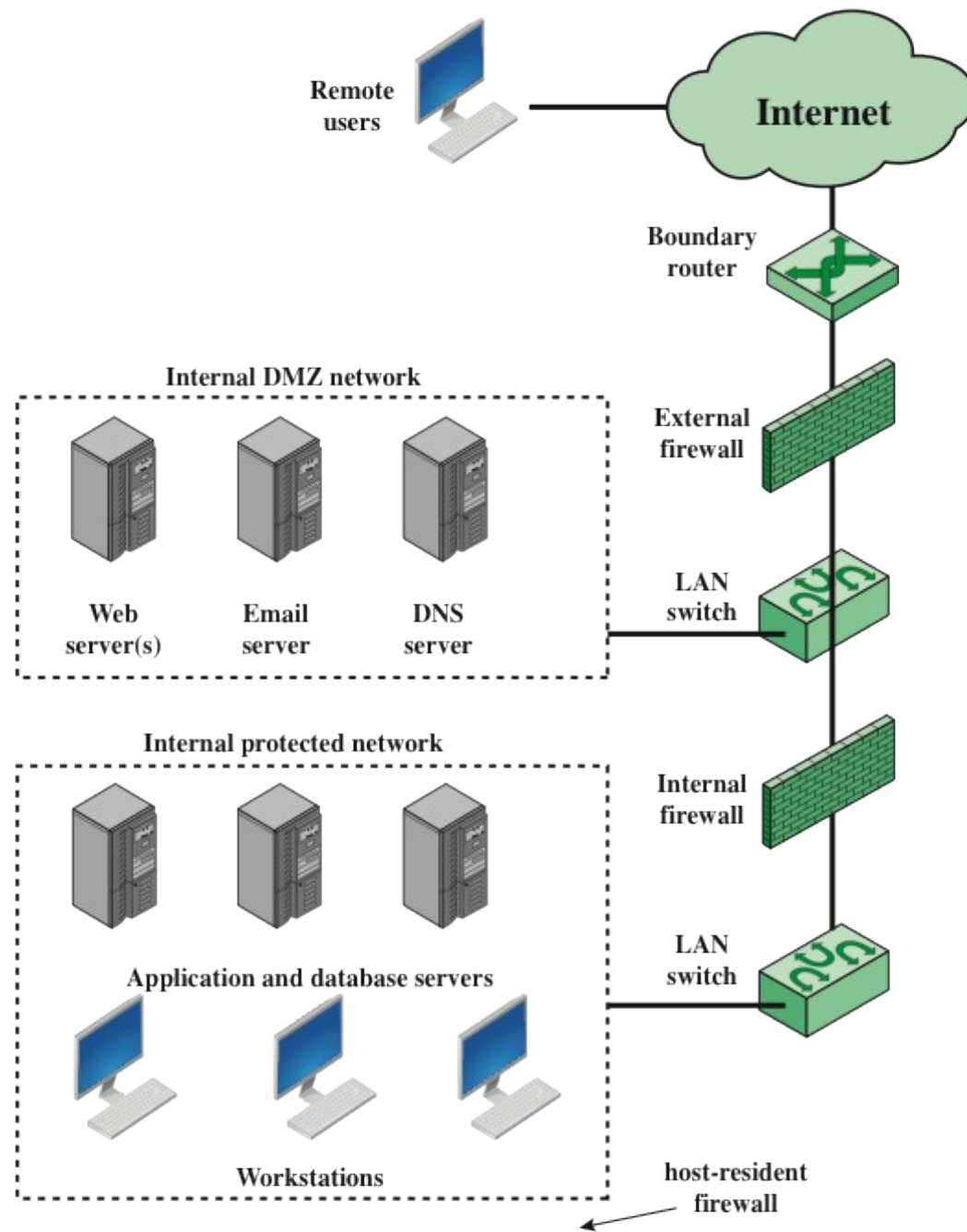


Figure 12.3 Example Firewall Configuration

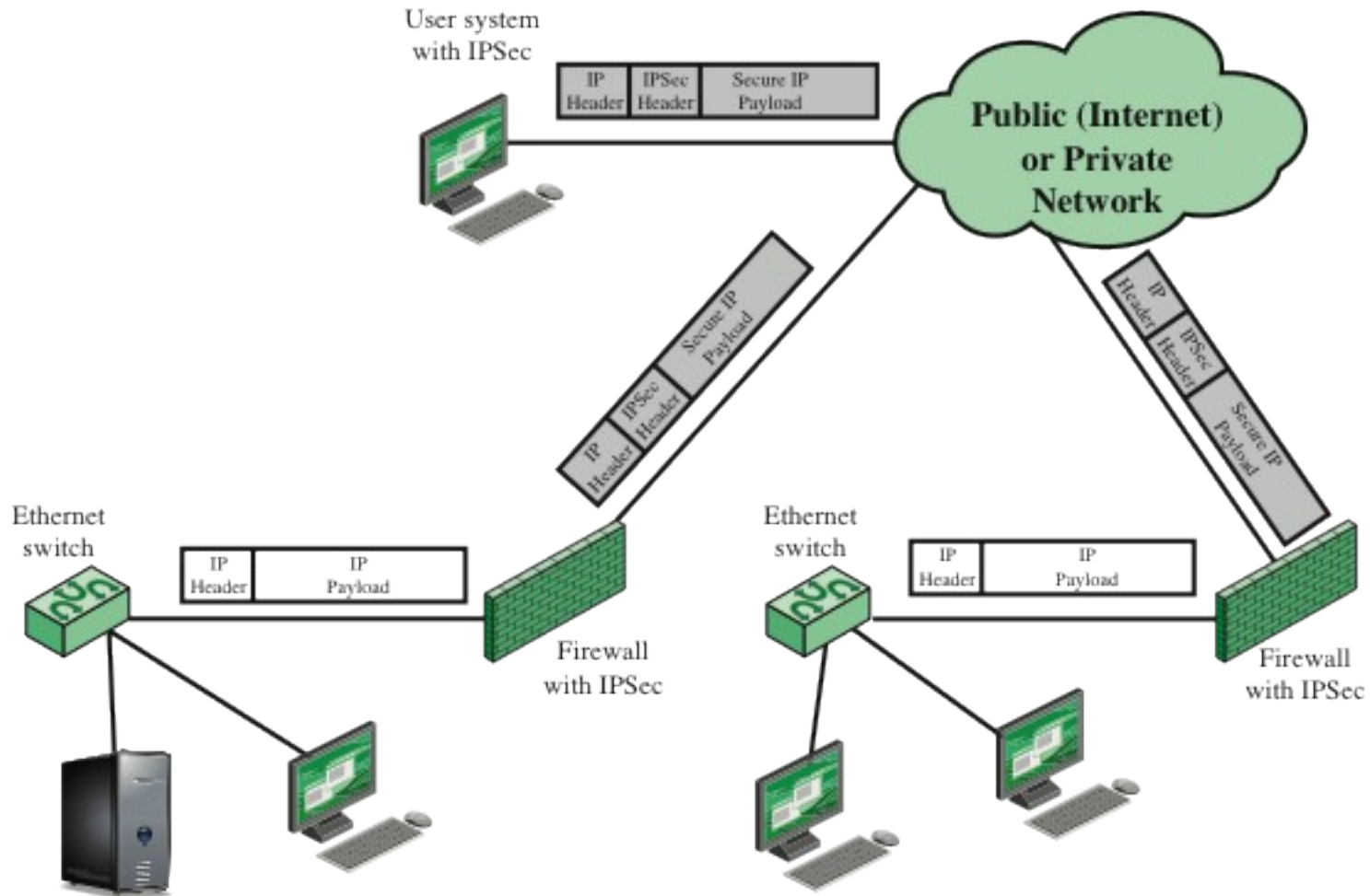


Figure 12.4 A VPN Security Scenario

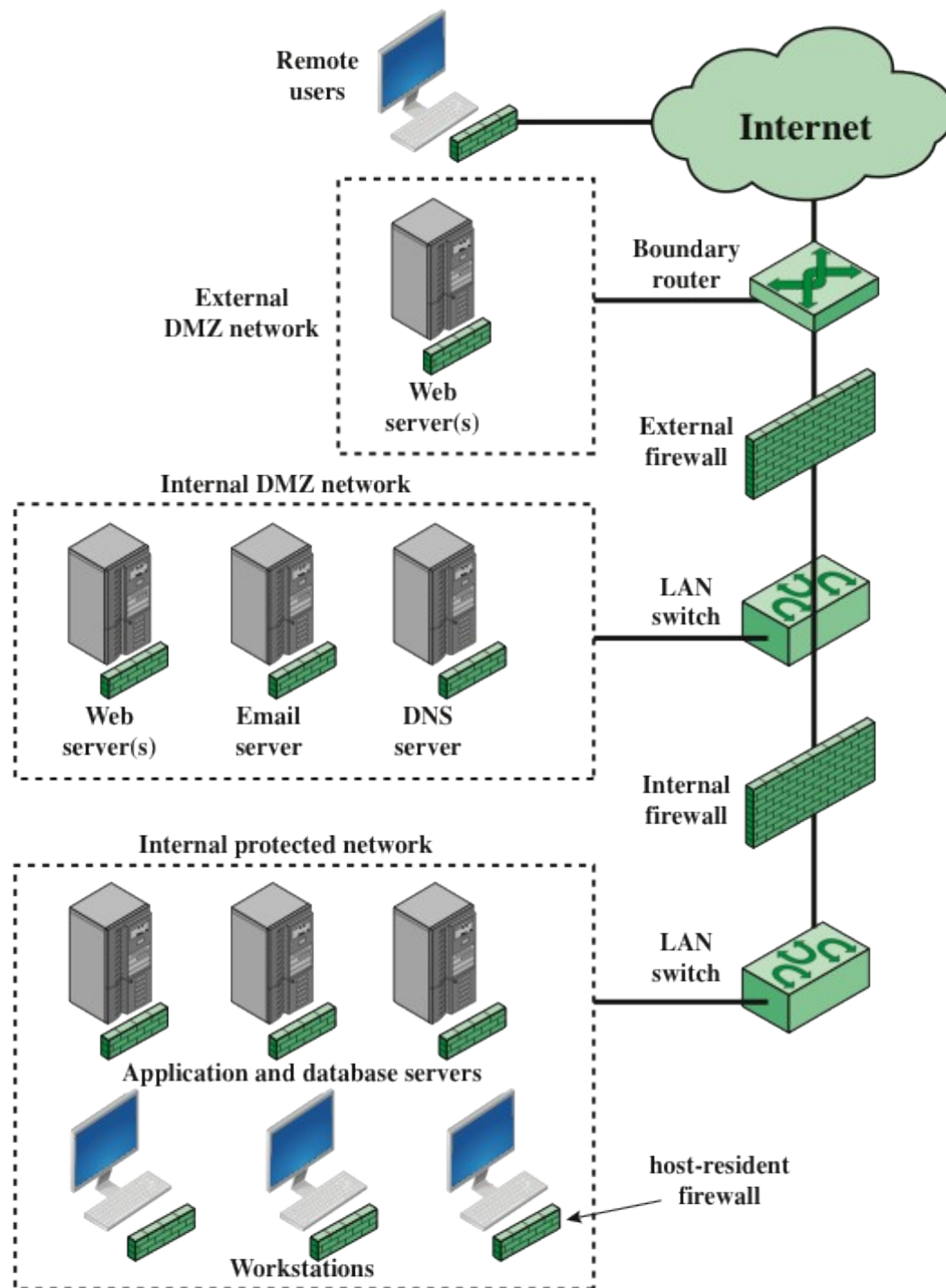


Figure 12.5 Example Distributed Firewall Configuration

Topologies

- **Host-resident firewall**

- This category includes personal firewall software and firewall software on servers
- Can be used alone or as part of an in-depth firewall deployment

- **Screening router**

- A single router between internal and external networks with stateless or full packet filtering
- This arrangement is typical for small office/home office (SOHO) applications

- **Single bastion inline**

- A single firewall device between an internal and external router
- This is the typical firewall appliance configuration for small-to-medium sized organizations

- **Single bastion T**

- Similar to single bastion inline but has a third network interface on bastion to a DMZ where externally visible servers are placed

- **Double bastion inline**

- DMZ is sandwiched between bastion firewalls

- **Double bastion T**

- DMZ is on a separate network interface on the bastion firewall

- **Distributed firewall configuration**

- Used by some large businesses and government organizations

Summary

- The need for firewalls
- Firewall characteristics
- Types of firewalls
 - Packet filtering firewall
 - Stateful inspection firewalls
 - Application level gateway
 - Circuit level gateway
- Firewall basing
 - Bastion host
 - Host based firewalls
 - Personal firewall
- Firewall locations and configurations
 - DMZ networks
 - Virtual private networks
 - Distributed firewalls
 - Firewall location and topologies summary

Operating System Security

General steps to follow for securing an OS

1. Disable all unnecessary services.
2. Restrict permissions on files and access to the Registry.
3. Remove unnecessary programs.
4. Apply the latest patches and fixes.

Operating System Security

- Services Like servers, many workstations also have the ability to enable and disable services.
- Services can be disabled through the Services administration tool on Windows platforms, by commenting the service out of `inetd.conf`, or by disabling it through the appropriate service file in `xinetd.conf` under UNIX.

Operating System Security

- File system Controlling access is an important element in maintaining system security.
- In practice, maintaining the least privileged principle directly affects the level of administrative, management, and auditing overhead, increasing the levels required to implement and maintain the environment.
- One alternative, the use of user groups, is a great time saver. Instead of assigning individual access controls, groups of similar users are assigned the same access. In cases where all users in a group have exactly the same access needs, this method works.
- However, in many cases, individual users need more or less access than other group members. When security is important, the extra effort to fine-tune individual user access provides greater control over what each user can and cannot access.

Operating System Security

- Removing unnecessary programs

The default installation of many operating systems includes programs that are unnecessary.

- For example, Microsoft updates are often specifically labeled Security Updates

Güvenlik İpuçları

- Yazılımların asıllarına uygunlukları testi için anahtar sunucuları kullanın (gpg, vb.)
- Gereksiz şekilde çalışan servislerin veya arka kapı yazılımların tespiti (netstat komudu)
- ARP Aldatmasının Tespit edilmesi (arpwatch komudu)
- Statik ARP Tablosu oluşturulması
- Netfilter, iptables examples

<http://www.thegeekstuff.com/2011/06/iptables-rules-examples/>

Iptables grafik arayüzleri

<http://www.iptables.info/en/iptables-gui.html>

Güvenlik İpuçları

- Port tarama (nmap, vb.)
- Ağınızdaki Açıkları Tarayın (Nessus yazılımı)
- Kendi (AAA) sertifika (X.509) merkezini oluşturun.
- Ağ trafik denetimi için argus vb. denetleme yazılımları kullanın.

Daha fazla ipucu için kitap önerisi

“Windows, Linux, *BSD için Ağ Güvenliği İpuçları”,
100 Etkin Güvenlik Tekniği, Yazar: Andrew Lockhart

Kaynaklar

- **Introduction to Networks (Cisco CCNA 1)**

http://139.179.33.220/CCNA5.0/CCNA5.0_Introduction_to_Networks/index.html

- **Cisco CCNA Security**
- Kitap “Network Security Essentials” Yazar: W. Stallings
<http://williamstallings.com/NetworkSecurity/>