HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK

Bevezetés az IT biztonságba

VIHIBB01 – Kódolás és IT biztonság (2023)

Dr. Buttyán Levente

CrySyS Lab, BME
buttyan@crysys.hu

www.crysys.hu

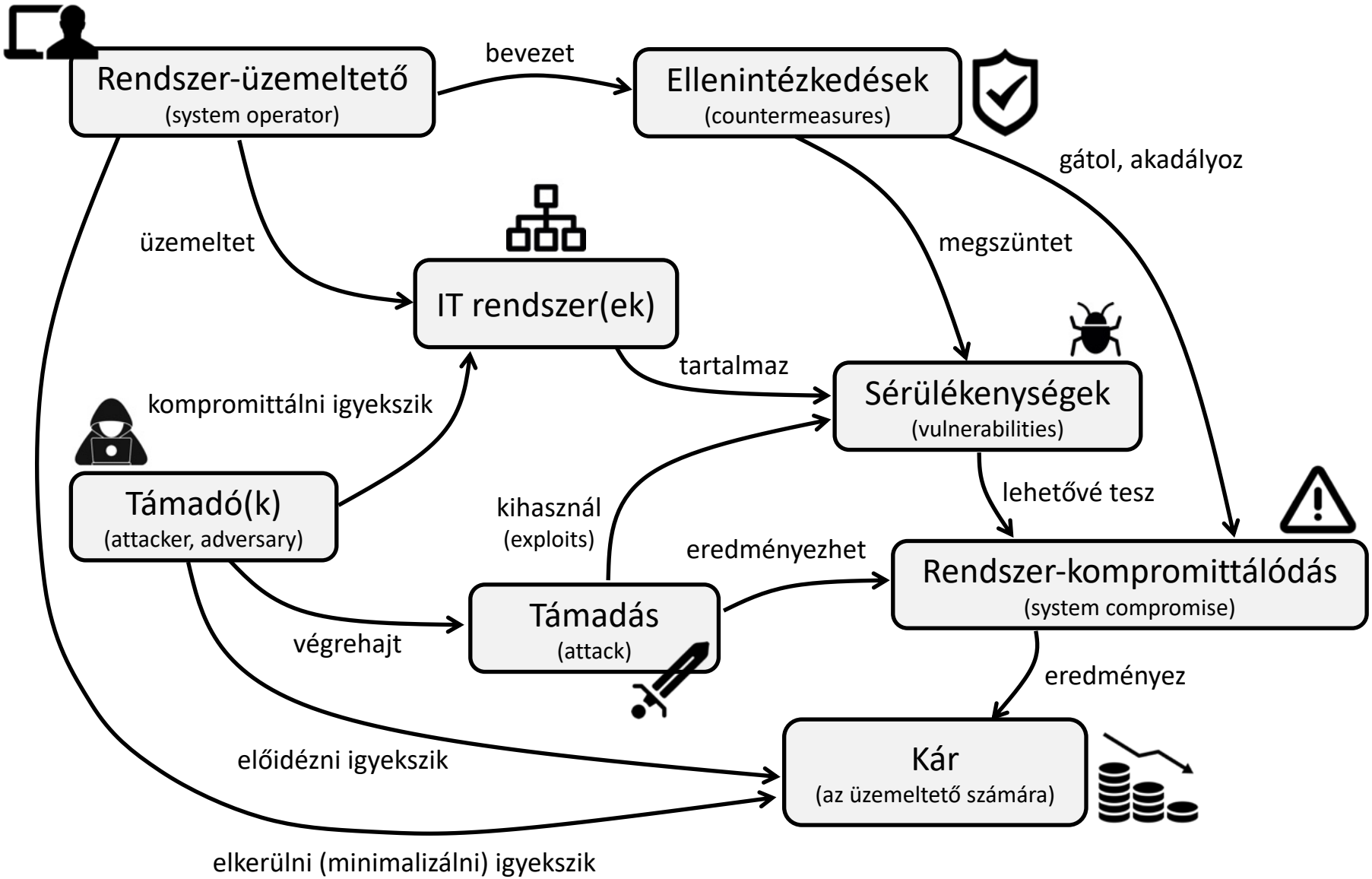
MŰEGYETEM 1782

Tartalom

- IT biztonsági fogalmak és összefüggések bevezetése
- A biztonsági kockázatot befolyásoló tényezők áttekintése
  - Jellemző támadói profilok, modellek
  - Sérülékenységek, kihasználható gyengeségek
  - Biztonsági mechanizmusok, ellenintézkedések
- A biztonsági incidenskezelés rövid áttekintése

Mivel foglalkozik az IT biztonság?

Színtér, szereplők, alapkonfliktus



A kompromittálódás fajtái

- A rendszer szolgáltatásaihoz vagy erőforrásaihoz történő illetéktelen hozzáférés, azok illegitim módon történő használata, módosítása, vagy elérhetetlenné tétele

  Példák:

  – Illegitim belépés egy legitim felhasználó fiókjába
  – Számítógép kártékony programmal (malware) történő megfertőzése
  – Kiszolgáló (szerver) elárasztása nagy mennyiségű kéréssel, ami annyira leterheli, hogy nem tud legitim kéréseket kiszolgálni (Denial-of-Service, DoS)

- A rendszerben tárolt, feldolgozott, vagy továbbított információ bizalmasságának, integritásának, vagy rendelkezésre állásának sérülése

  Példák:

  – Jelszó vagy üzleti titok kiszivárgása
  – Adatbázisban tárolt adatok illegitim módosítása
  – Háttértáron tárolt adatok zsaroló vírus által történő rejtjelezése

Szándékos vs. véletlen

- A kompromittálódás mindig valamilyen szándékos tevékenység (támadás) eredménye
- A véletlen hibákból, természeti katasztrófákból származó nem kívánatos állapot (failure) nem kompromittálódás

  Példák:

  – A háttértár fizikai meghibásodása miatt az adatok elérhetetlenek
  – Kommunikációs hiba miatt egy üzenet tartalma módosul
  – A rendszer tűzvészben fizikailag megsemmisül
- Bár az eredményük hasonló lehet, a szándékos támadások és a véletlen hibák kezelése különböző technikákat igényel
  – szándékos támadások → biztonság (security), megbízhatóság (trustworthiness)
  – véletlen hibák → hibatűrés (fault tolerance), megbízhatóság (reliability), (üzem)biztonság (safety)

A CIA triád

C = Confidentiality (bizalmasság)

- az **információhoz** történő jogosulatlan hozzáférés megakadályozása

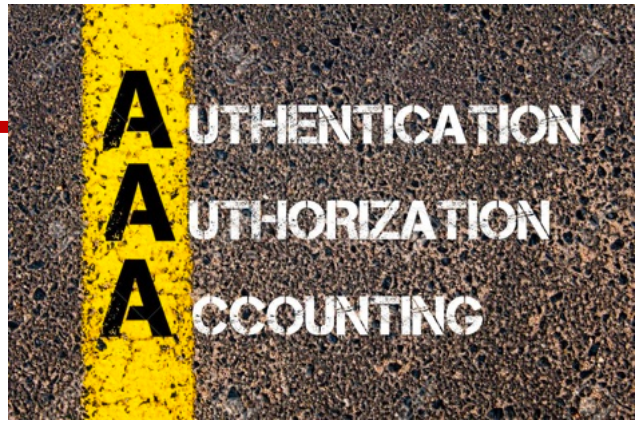
I = Integrity (integritás)

- az **információ** jogosulatlan módosításának megakadályozása

A = Availability (rendelkezésre állás)

- az **információ** elérhetőségének biztosítása az arra jogosultak számára

AAA



Authentication (hitelesítés)

- szolgáltatáshoz vagy erőforráshoz történő hozzáférést kezdeményező identitásának ellenőrzése (ki akar hozzáférni?)

Authorization, access control (engedélyezés, hozzáférés-vezérlés)

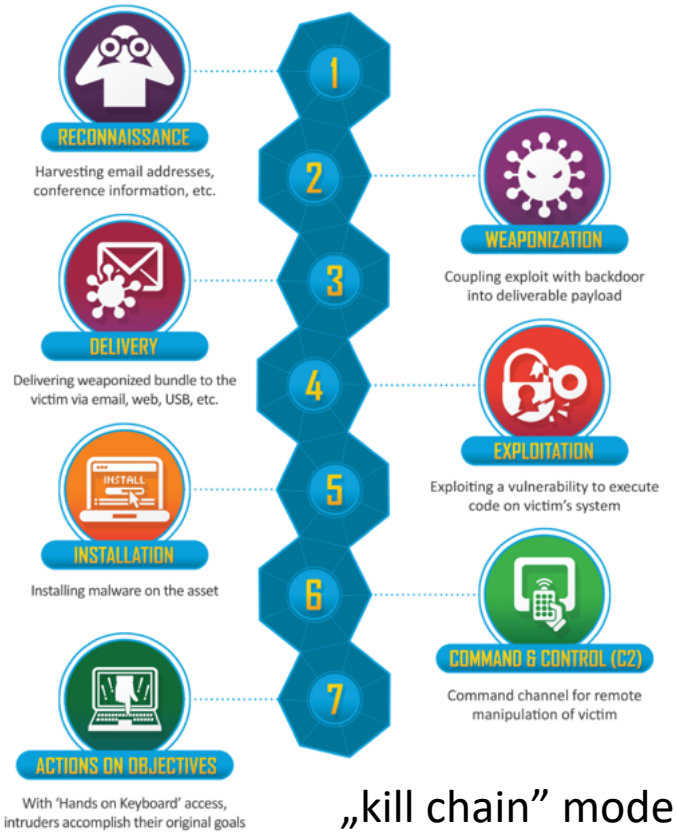
- szolgáltatáshoz vagy erőforráshoz történő hozzáférés engedélyezése
- függhet a hozzáférést kezdeményező identitásától, a hozzáférés jellegétől (pl. írás, olvasás), és egyéb körülményektől (pl. idő)

Accounting (felelősségre vonhatóság)

- a rendszerhez történt hozzáférések, az elvégzett műveletek utólagos visszakereshetőségének, ellenőrizhetőségének, a hozzáférést végző entitás azonosíthatóságának és felelősségre vonhatóságának lehetősége

Támadás

- A támadás az a folyamat vagy tevékenység, mely során a támadó kompromittálja a támadott rendszert, annak sérülékenységeit kihasználva
- A támadás általában egy komplex folyamat/tevékenység
- A támadás felépítésének, az alkalmazott technikáknak az ismerete elengedhetetlen a támadás megakadályozását célzó biztonsági mechanizmusok tervezéséhez

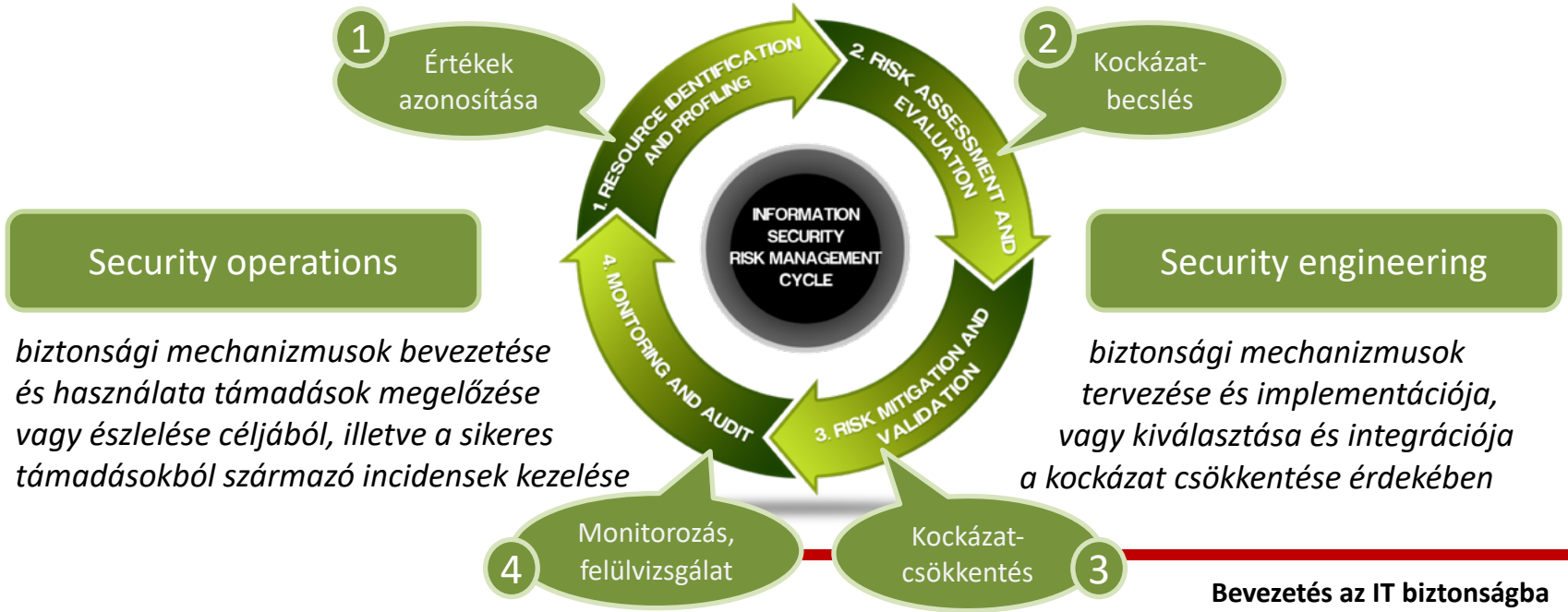


„kill chain" modell

Az üzemeltető nézőpontja

biztonság = kockázat (risk) menedzsment

- a kockázat a sikeres támadásból eredő veszteség várható értéke
- a kockázat általában nem eliminálható teljesen, mert minden elképzelhető támadás megakadályozása túl drága (még ha egyáltalán lehetséges is volna)
- az üzemeltető célja tehát a kockázat minimalizálása adott költségvetés (budget) mellett
- ezt nevezzük kockázat-menedzsmentnek, ami egy ciklikus folyamat...



A kockázatot befolyásoló tényezők

Risk = Likelihood x Impact
(of attacks)

- Impact:
  - a sikeres támadásból bekövetkező potenciális veszteség
    - » közvetlen veszteség (pl. bevétel kiesés, a helyreállítás költségei)
    - » közvetett veszteség (pl. jó hírnév elvesztése, bizalom csökkenése)

- Likelihood:
  - a sikeres támadás valószínűsége, mely függ
    - » a támadótól (motiváció, szándék, lehetőség, képesség, erőforrások)
    - » a rendszer sérülékenységeitől (kihasználható gyengeségek)
    - » az alkalmazott ellenintézkedésektől (biztonsági mechanizmusok)

„ … ha ismerjük az ellenséget és ismerjük magunkat is, akkor száz csatában sem jutunk veszedelembe; ha azonban nem ismerjük az ellenséget, csak magunkat ismerjük, akkor egyszer győzünk, másszor vereséget szenvedünk; és ha sem az ellenséget, sem magunkat nem ismerjük, akkor minden egyes csatában feltétlenül végveszély fenyeget bennünket.”

— Szun Ce, *A háború művészete*

A támadó

A támadó jellemzői

- Motiváció + Szándék (intent) + Lehetőség (opportunity)
- Képességek
  - Információszerző képesség
  - Műszaki tudás, szakértelem
  - Megtévesztési képességek
- Erőforrások

Motivációk

- Gazdasági haszonszerzés
- Politikai célok elérése
- Társadalmi üzenet közvetítése
- Vallási üzenet közvetítése, cél elérése
- Bosszú
- Erő, képesség demonstrációja
- ...

Információszerző képesség

- A támadás sikere nagy mértékben függ attól, hogy mennyi és milyen jellegű információval rendelkezik a támadó a megtámadott rendszerről
- Az információszerzés megelőzheti magát a támadást, és folytatódhat a támadás alatt is
- A támadó számára hasznos információk:
  - a rendszer általános felépítése, elérhető szolgáltatásai, hardver és szoftver komponensei és azok konfigurációja, a hálózati topológia, az alkalmazott protokollok, ...
  - az alkalmazott biztonsági mechanizmusok (pl. tűzfalak, behatolás detektáló rendszerek, anti-vírus szoftverek, ...)
  - a rendszer és az alkalmazott biztonsági megoldások ismert sérülékenységei
  - kik a rendszer felhasználói, milyen jogosultságokkal rendelkeznek?
  - ...

Műszaki tudás, szakértelem

- A támadó szakértői tudása segítségével transzformálja a megszerzett információt sikeres támadássá
- A műszaki tudás, szakértelem az információszerzést is segítheti
- A támadói szakértelem szintjei:
  - alapvető műszaki és informatikai ismeretek (hardver; operációs rendszerek; hálózati technológiák, protokollok; elosztott rendszerek; alkalmazások; ...)
  - ismert sérülékenységek és támadási módszerek, eszközök ismerete; IT biztonsági szakértelem
  - új sérülékenységek felfedezésének, új támadási módszerek és eszközök kifejlesztésének képessége

Erőforrások

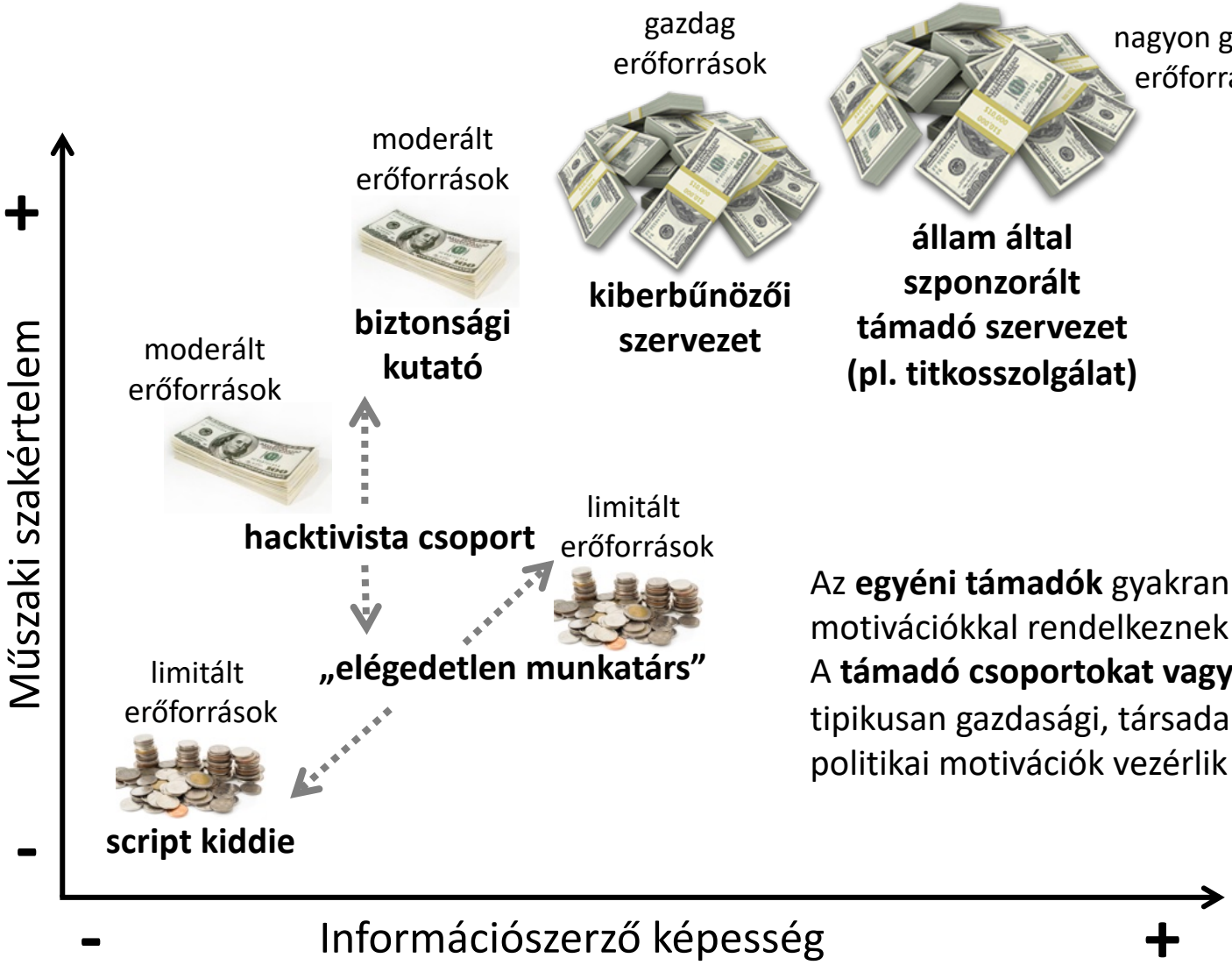
erőforrás = pénz

- szabadon konvertálható információvá, tudássá, szakértelemmé, emberi erőforrássá, ...

Példák:

- Információszerző képesség növelése
  - » megvesztegetés, zsarolás
  - » műszaki dokumentációk megvásárlása
  - » kifinomult megtévesztési módszerek (social engineering) támogatása
  - » fejlett OSINT, SIGINT módszerek alkalmazásának lehetősége
- Műszaki tudás, szakértelem növelése
  - » szakértők szerződtetése
  - » saját kompetenciák növelése képzéssel, tréninggel
- Speciális támadói képességek növelése
  - » 0-day sérülékenységeket kihasználó exploit-ok vásárlása
  - » fejlett kripto-analízis eszközök használata (kódfeltörés, hamisítás)
  - » nagy számítási kapacitás bérlése

Tipikus támadó profilok (modellek)



Az **egyéni támadók** gyakran személyes motivációkkal rendelkeznek
A **támadó csoportokat vagy szervezeteket** tipikusan gazdasági, társadalmi, vagy politikai motivációk vezérlik

Kiberbűnözői szervezet

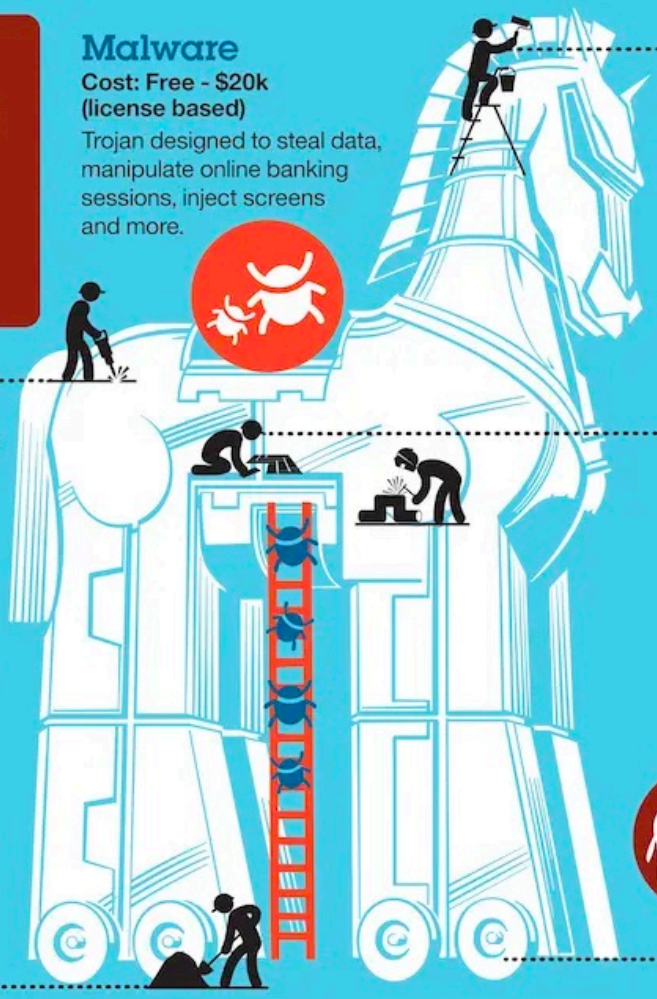
- Legnagyobb probléma ma egy átlagos felhasználó vagy cég számára
- Motiváció: gazdasági haszonszerzés
- Információszerző képesség: **fejlett**
  - műszaki megoldások (spyware telepítése; szerverek, felhasználói fiókok feltörése)
  - megtévesztés (phishing, social engineering)
- Műszaki tudás, szakértelem: **fejlett**
- Erőforrások: **gazdag**
  - „szakértők, specialisták” alkalmazása
  - információk és eszközök (pl. exploit-ok, malware) vásárlása
  - komoly háttér infrastruktúra fenntartása
  - térben és időben kiterjedt támadó kampányok futtatása

Cybercriminal Ecosystem

IBM

Cybercrime is no longer a one man operation. Within the cybercrime underground an attacker can find a wealth of tools and services that can be bought or rented to facilitate different aspects of the attack lifecycle.*

Fraud as a service is constantly changing and adapting to new security solutions, offering end to end technologies, multiple SLA levels and low prices for everything a cybercriminal might need.



Malware

Cost: Free - $20k (license based)

Trojan designed to steal data, manipulate online banking sessions, inject screens and more.

Exploit Kits

Cost: $2K (monthly rental)

Toolkits designed to exploit system and software vulnerabilities resulting in a malicious download.

Infrastructure

Cost: $50 - $1,000 (Rental per month)

Hosting services for malware update, configuration and command and control servers. Some are fast flux or TOR based.

Droppers

Cost: Free - $10K

Software designed to download malware to an infected device, evading antivirus and research tools.

Spammers

Cost: $1 - $4 per 1000 emails

Spam botnet operators that spread emails with attachments or links leading to a Trojan infection.

Money Mules

Cost: Up to 60% of account balance

A person who receives thestolen money from a hacked account and transfers the funds via an anonymous payment service to the mule operator.

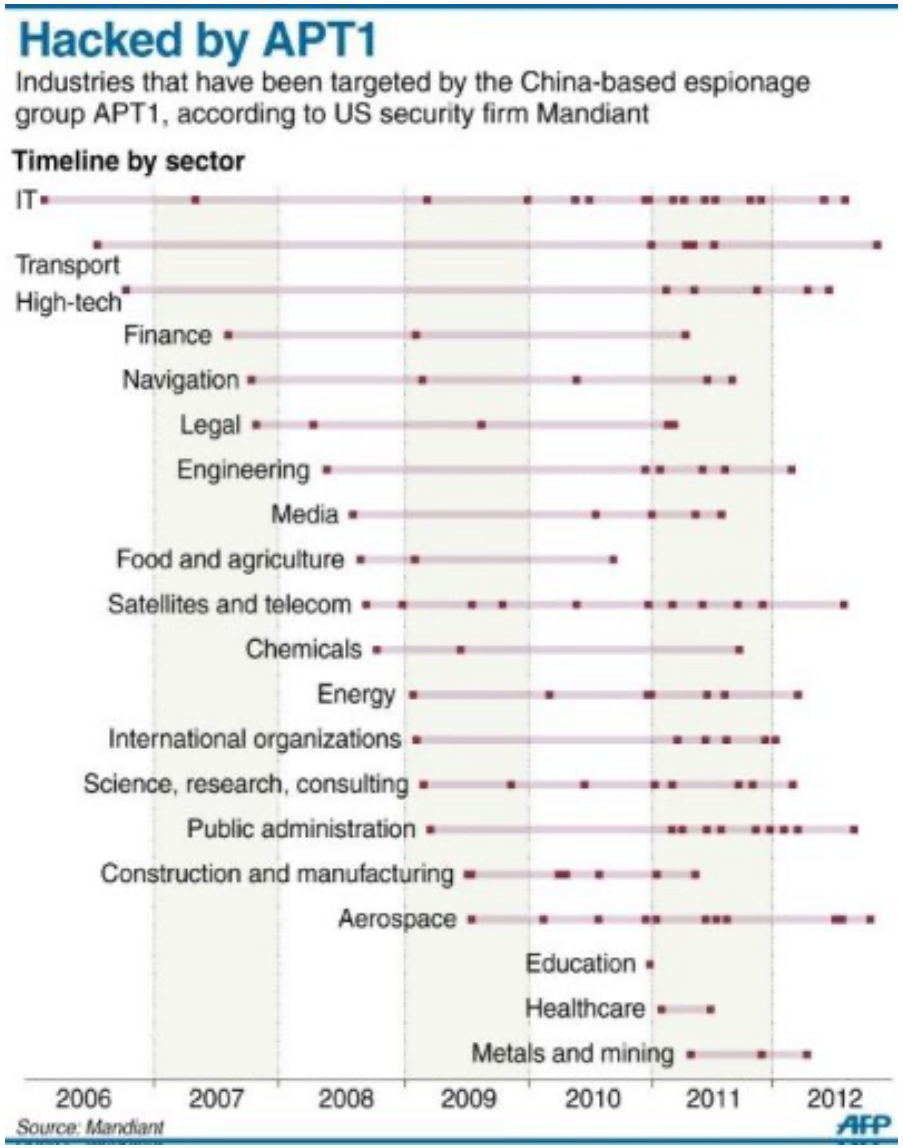
* This infographic shows one possible scenario of a cybercriminal attack lifecycle. Prices for this scenario are estimates.

Állam által szponzorált támadó szervezet

- Legnagyobb probléma ma kormányzatok és bizonyos cégek számára
- Motiváció: politikai vagy gazdasági célok elérése
  - világos stratégiai célok (kémkedés vagy szabotázs)
- Információgyűjtő képesség: **nagyon fejlett**
  - fejlett megfigyelő (surveillance) eszközök (pl. telefonra telepítve)
  - „hagyományos” lehallgatás, információszerzés (SIGINT)
- Műszaki tudás, szakértelem: **nagyon fejlett**
  - komplex K+F és tréning programok
- Erőforrások: **nagyon gazdag**
  - „szakértők, specialisták” alkalmazása vagy kiképezése
  - információ és eszközök (pl. exploit-ok, malware) vásárlása
  - komoly háttér infrastruktúra fenntartása
  - körültekintő tervezés, hosszú élettartamú, célzott támadások
- Példák:
  - PLA Unit 61398 (Kína)
  - TAO (USA)

PLA Unit 61398 (aka APT1)

- Közel 150 áldozat 7 év alatt
- Átlagosan 356 napig volt bent az áldozatok rendszerében
- Az azonosított székhely mérete a szervezet méretét is jelzi: potenciálisan több száz fő!



Office of Tailored Access Operations (TAO)

- „a számítógépes hadviselés egyik információgyűjtő egysége az NSA-n belül”
- „feladata a külföldi szervezetek által működtetett számítógépes hálózatok azonosítása, megfigyelése, az azokba való behatolás és azokból információszerzés.”
- „az adott hálózat védelmi rendszerét, testre szabott, az adott feladatra koncentráló módszerrel törik fel”
- „több mint 1000 katonai és polgári hackert, információ-elemzőt, és más számítógépes szakembereket foglalkoztat”
- „2013-ig valószínűleg összesen 85 ezer számítógépet fertőzött meg, közöttük demokratikusan megválasztott kormányok számítógépes rendszereit is”

Forrás: https://hu.wikipedia.org/wiki/Tailored_Access_Operations

Célzott támadások

- célzott = az áldozat kiszemelt, nem véletlen választott
  - az áldozat lehet egy cég, szervezet, egyén, vagy egyének kisebb csoportja
- Testreszabott támadó eszközök és behatolási technikák
  - spear phishing vagy fejlett social engineering
  - támadás szerződéses partneren, beszállítón keresztül
  - több különböző exploit alkalmazása (gyakran 0-day vagy nagyon friss)
- Időben kiterjedt (persistent), mégis rejtve maradó (stealthy) működés
  - anti-vírus és behatolás detektáló eszközök kijátszása
  - körültekintő tervezés és tesztelés
- Erőforrásokban gazdag háttér
  - katonai egységek, titkosszolgálatok
  - nagy cégek (célzottan versenytársakat támadnak)

Stuxnet (2010. június)

- “the Most Menacing Malware in History” (Kim Zetter, Wired)
- Cél: natanz-i uránium dúsító üzem Iránban
- Forrás: USA és/vagy Izrael (feltételezés)
- Motiváció: Irán atomprogramjának szabotálása (képesség demonstrációja)
- kifinomult malware, több 0-day exploit, hamis digitális aláírás

WinCC PLC menedzsment szoftvert futtató PC

A Stuxnet megfertőzte a PC-t, átvette a PC és a PLC-k közötti kommunikáció irányítását, módosította az üzeneteket, ...

Az uránium centrifugák forgását vezérlő PLC

... és átprogramozta a PLC-ket. A módosított program helytelenül vezérelte a centrifugák forgását.

Uránium dúsító centrifigák

A helytelen vezérlés fizikailag tönkretette a centrifugákat.

Duqu (2011. október)

Home / News & Blogs / Zero Day

Hungarian Lab found Stuxnet-like Duqu malware

By Ryan Naraine | October 21, 2011, 9:11am PDT

Summary: *The Laboratory of Cryptography and System Security (CrySyS) in Hungary confirmed its participation in the initial discovery of the Duqu cyber-surveillance Trojan.*

A security lab attached to the Budapest University of Technology and Economics in Hungary has come forward as the mystery outfit that found the Stuxnet-like "Duqu" cyber-surveillance Trojan.

According to Symantec's initial report on Duqu [PDF], the malware sample was passed along by an unnamed "research lab with strong international connections," a statement that led to speculation about the origins and intent of the threat.

Néhány aktuális példa...

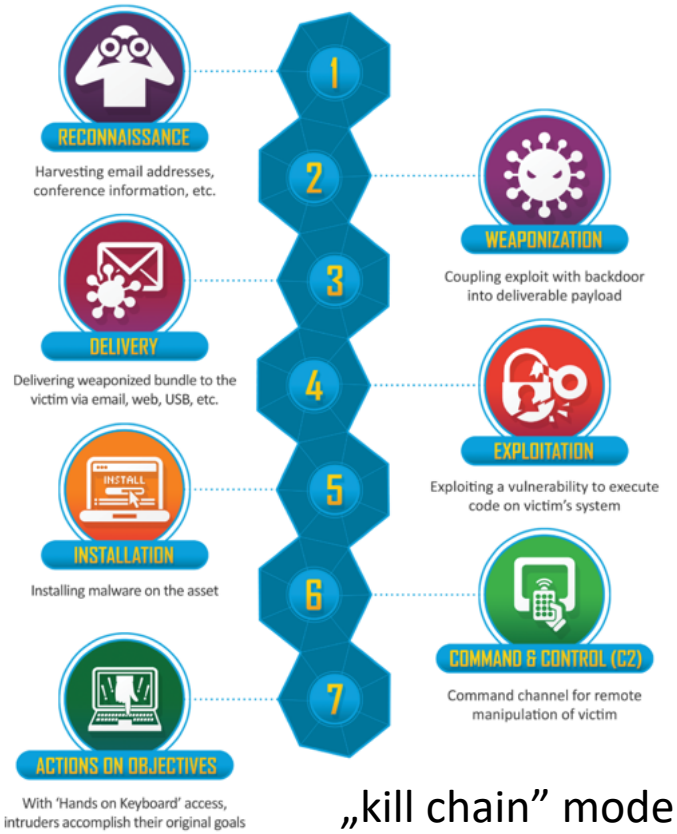
Forrás: https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence

További források:

https://www.talosintelligence.com/
https://securelist.com/threat-category/apt-targeted-attacks/
https://www.mandiant.com/resources/insights/apt-groups

Támadás (emlékeztető)

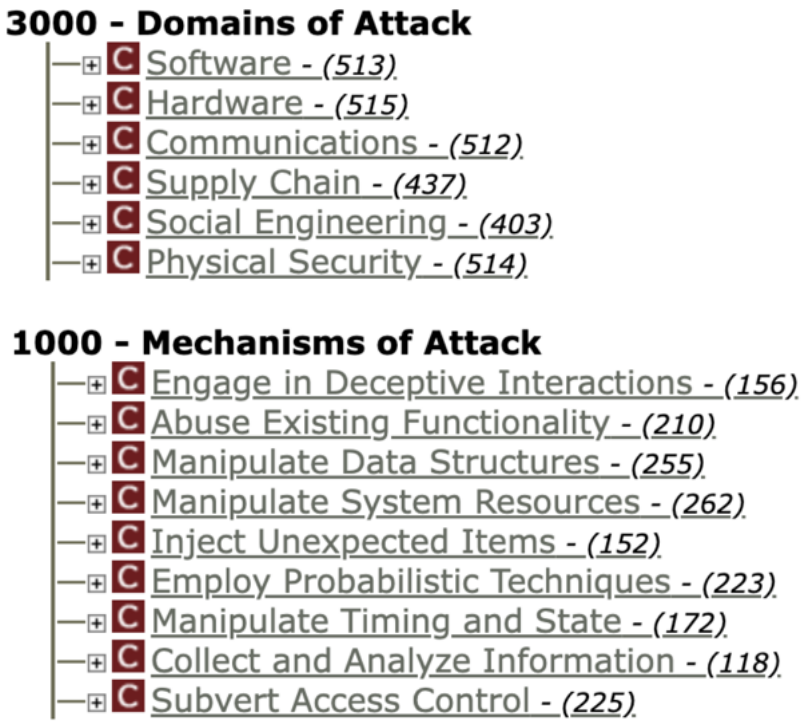
- A támadás az a folyamat vagy tevékenység, mely során a támadó kompromittálja a támadott rendszert, annak sérülékenységeit kihasználva
- A támadás általában egy komplex folyamat/tevékenység
- A támadás felépítésének, az alkalmazott technikáknak az ismerete elengedhetetlen a támadás megakadályozását célzó biztonsági mechanizmusok tervezéséhez



„kill chain" modell

CAPEC

- Common Attack Pattern Enumeration and Classification
  - Ismert támadási minták átfogó gyűjteménye
  - Közösségi erőforrás, szabadon elérhető on-line: **https://capec.mitre.org/**
  - 559 támadási minta különböző szempontok (pl. alkalmazott támadási mechanizmusok) szerint rendezve



CAPEC – példa

Software → Software Integrity Attacks →

CAPEC-185: Malicious Software Download

Attack Pattern ID: 185
Abstraction: Standard

View customized information: Conceptual | Operational | Mapping-Friendly | **Complete**

Description

An attacker uses deceptive methods to cause a user or an automated process to download and install dangerous code that originates from an attacker controlled source. There are several variations to this strategy of attack.

Typical Severity

Very High

Relationships

Nature	Type	ID	Name
ChildOf	M	184	Software Integrity Attack
CanFollow	S	159	Redirect Access to Libraries
CanPrecede	S	662	Adversary in the Browser (AiTB)

View Name	Top Level Categories
Domains of Attack	Software
Mechanisms of Attack	Manipulate System Resources
Supply Chain Risks	Sustainment

Related Weaknesses

CWE-ID	Weakness Name
494	Download of Code Without Integrity Check

Sérülékenységek

A sérülékenységek fajtái

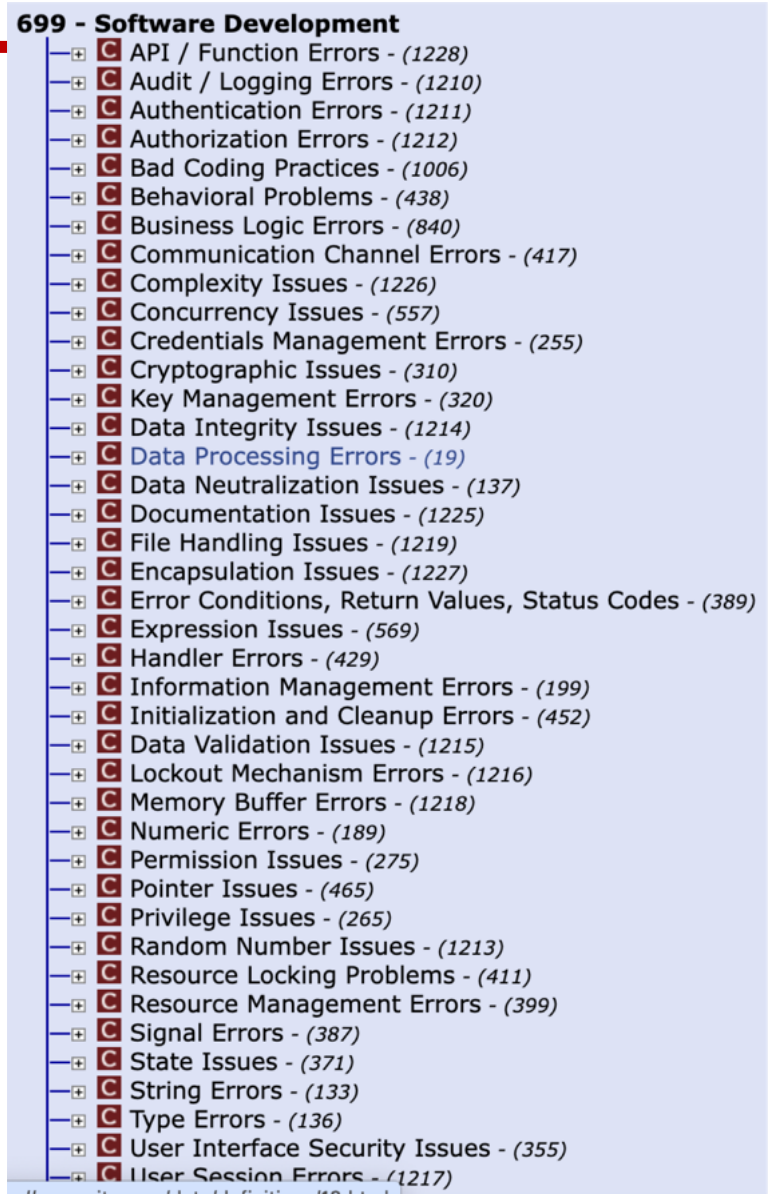
- **Műszaki (logikai) –** tervezési és implementációs hibák a rendszert alkotó hardver és szoftver komponensekben, a komponensek közötti interfészekben, a komponensek között használt protokollokban
- **Fizikai –** olyan gyengeségek, melyek kihasználása fizikai hozzáférést eredményezhet a rendszerhez, vagy annak valamely komponenséhez (pl. folyosón elhelyezett router)
- **Üzemeltetési –** kihasználható gyengeségek a rendszer üzemeltetése során alkalmazott eljárásokban (pl. alapértelmezett jelszó megváltoztatása nincs kikényszerítve)
- **Személyi –** az üzemeltető személyzetet, alkalmazottakat, szerződéses partnereket érintő gyengeségek (pl. biztonságtudatosság hiánya, szakértelem hiánya, lefizethetőség, megzsarolhatóság)

A sérülékenységek okai

- A rendszerek nagy komplexitása
  - ami bonyolult, abban több a hibalehetőség
- Alkalmas tervezési, implementációs, ellenőrzési és tesztelési módszerek hiánya vagy hiányosságai
  - pl. a javasolt formális ellenőrzési módszerek nem skálázódnak
  - pl. a tesztelés sosem lehet kimerítő a gyakorlatban használt rendszerméretek és –komplexitás mellett
- Korlátozott erőforrások
  - pénz, idő, szakértelemmel bíró emberi munkaerő
- Rossz feltételezések a tervezés vagy az üzemeltetés során
  - pl. egy adott támadó típus, támadási módszer figyelmen kívül hagyása
- Gyenge minőségű specifikáció az implementáció számára
  - következményként a szakértő tervező helyett a kevésbé hozzáértő programozó hoz tervezési döntéseket
- (Hanyagság a tervező, kivitelező, üzemeltető oldalán...)

CWE

- Common Weakness Enumeration
  - Ismert szoftver és hardver gyengeségek átfogó gyűjteménye
  - Közösségi erőforrás, szabadon elérhető on-line: **https://cwe.mitre.org/**
  - 933 gyengeség különböző szempontok (pl. szoftver, hardver, kutatás) szerint rendezve



CWE – példa

Research concepts → Protection Mechanism Failure → Insufficient Verification of Data Authenticity →

CWE-494: Download of Code Without Integrity Check

Weakness ID: 494
Abstraction: Base
Structure: Simple

View customized information: Conceptual | Operational | Mapping Friendly | **Complete** | Custom

Description

The product downloads source code or an executable from a remote location and executes the code without sufficiently verifying the origin and integrity of the code.

Extended Description

An attacker can execute malicious code by compromising the host server, performing DNS spoofing, or modifying the code in transit.

Relationships

Modes Of Introduction

Phase	Note
Architecture and Design Implementation	OMISSION: This weakness is caused by missing a security tactic during the architecture and design phase.

Applicable Platforms

Languages

Class: Not Language-Specific *(Undetermined Prevalence)*

Common Consequences

Scope	Impact	Likelihood
Integrity Availability Confidentiality Other	**Technical Impact:** *Execute Unauthorized Code or Commands; Alter Execution Logic; Other* Executing untrusted code could compromise the control flow of the program. The untrusted code could execute attacker-controlled commands, read or modify sensitive resources, or prevent the software from functioning correctly for legitimate users.	

Likelihood Of Exploit

Medium

CWE – példa

Demonstrative Examples

Example 1

This example loads an external class from a local subdirectory.

Example Language: **Java** *(bad code)*

```
URL[] classURLs= new URL[]{
   new URL("file:subdir/")
};
URLClassLoader loader = new URLClassLoader(classURLs);
Class loadedClass = Class.forName("loadMe", true, loader);
```

This code does not ensure that the class loaded is the intended one, for example by verifying the class's checksum. An attacker may be able to modify the class file to execute malicious code.

Example 2

This code includes an external script to get database credentials, then authenticates a user against the database, allowing access to the application.

Example Language: **PHP** *(bad code)*

```
//assume the password is already encrypted, avoiding CWE-312

function authenticate($username,$password){

   include("http://external.example.com/dbInfo.php");

   //dbInfo.php makes $dbhost, $dbuser, $dbpass, $dbname available
   mysql_connect($dbhost, $dbuser, $dbpass) or die ('Error connecting to mysql');
   mysql_select_db($dbname);
   $query = 'Select * from users where username='.$username.' And password='.$password;
   $result = mysql_query($query);

   if(mysql_numrows($result) == 1){
      mysql_close();
      return true;
   }
   else{
      mysql_close();
      return false;
   }

}
```

CWE – példa

Potential Mitigations

Phase: Implementation

Perform proper forward and reverse DNS lookups to detect DNS spoofing.
Note: This is only a partial solution since it will not prevent your code from being modified on the hosting site or in transit.

Phases: Architecture and Design; Operation

Encrypt the code with a reliable encryption scheme before transmitting.

This will only be a partial solution, since it will not detect DNS spoofing and it will not prevent your code from being modified on the hosting site.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

Speficially, it may be helpful to use tools or frameworks to perform integrity checking on the transmitted code.

- When providing the code that is to be downloaded, such as for automatic updates of the software, then use cryptographic signatures for the code and modify the download clients to verify the signatures. Ensure that the implementation does not contain CWE-295, CWE-320, CWE-347, and related weaknesses.
- Use code signing technologies such as Authenticode. See references [REF-454] [REF-455] [REF-456].

Phases: Architecture and Design; Operation

Strategy: Environment Hardening

Run your code using the lowest privileges that are required to accomplish the necessary tasks [REF-76]. If possible, create isolated accounts with limited privileges that are only used for a single task. That way, a successful attack will not immediately give the attacker access to the rest of the software or its environment. For example, database applications rarely need to run as the database administrator, especially in day-to-day operations.

Phases: Architecture and Design; Operation

Strategy: Sandbox or Jail

Run the code in a "jail" or similar sandbox environment that enforces strict boundaries between the process and the operating system. This may effectively restrict which files can be accessed in a particular directory or which commands can be executed by the software.

OS-level examples include the Unix chroot jail, AppArmor, and SELinux. In general, managed code may provide some protection. For example, java.io.FilePermission in the Java SecurityManager allows the software to specify restrictions on file operations.

This may not be a feasible solution, and it only limits the impact to the operating system; the rest of the application may still be subject to compromise.

Be careful to avoid CWE-243 and other weaknesses related to jails.

Effectiveness: Limited

Note: The effectiveness of this mitigation depends on the prevention capabilities of the specific sandbox or jail being used and might only help to reduce the scope of an attack, such as restricting the attacker to certain system calls or limiting the portion of the file system that can be accessed.

CWE – példa

Detection Methods

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is typically required to find the behavior that triggers the download of code, and to determine whether integrity-checking methods are in use.

Note: These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules.

Black Box

Use monitoring tools that examine the software's process as it interacts with the operating system and the network. This technique is useful in cases when source code is unavailable, if the software was not developed by you, or if you want to verify that the build phase did not introduce any new weaknesses. Examples include debuggers that directly attach to the running process; system-call tracing utilities such as truss (Solaris) and strace (Linux); system activity monitors such as FileMon, RegMon, Process Monitor, and other Sysinternals utilities (Windows); and sniffers and protocol analyzers that monitor network traffic.

Attach the monitor to the process and also sniff the network connection. Trigger features related to product updates or plugin installation, which is likely to force a code download. Monitor when files are downloaded and separately executed, or if they are otherwise read back into the process. Look for evidence of cryptographic library calls that use integrity checking.

Automated Static Analysis

Automated static analysis, commonly referred to as Static Application Security Testing (SAST), can find some instances of this weakness by analyzing source code (or binary/compiled code) without having to execute it. Typically, this is done by building a model of data flow and control flow, then searching for potentially-vulnerable patterns that connect "sources" (origins of input) with "sinks" (destinations where the data interacts with external components, a lower layer such as the OS, etc.)

Effectiveness: High

Memberships

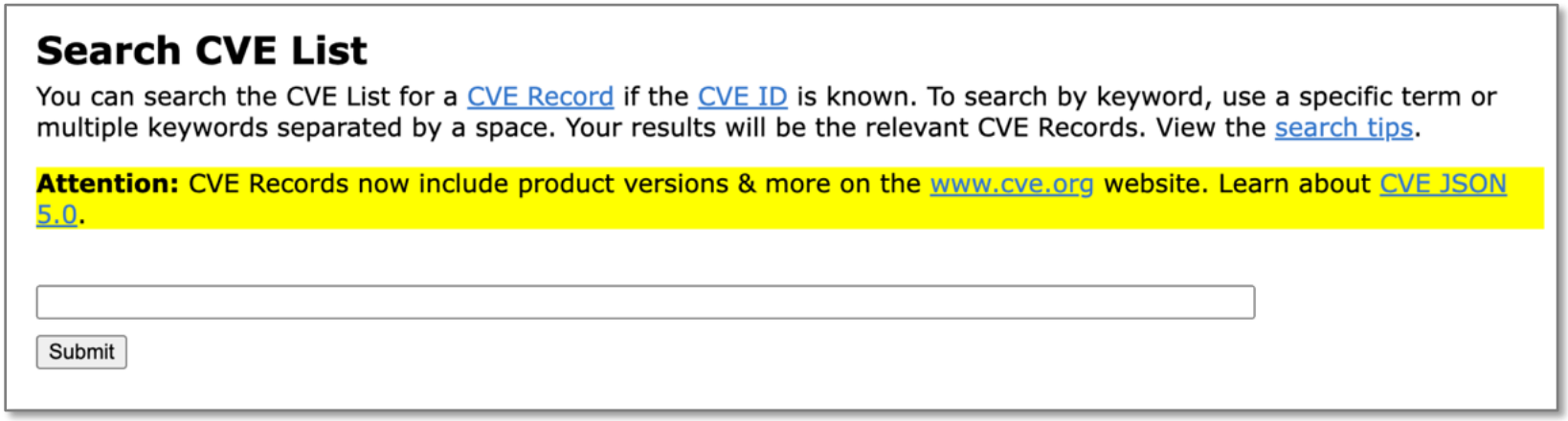
Nature	Type	ID	Name
MemberOf	C	752	2009 Top 25 - Risky Resource Management
MemberOf	C	802	2010 Top 25 - Risky Resource Management
MemberOf	C	859	The CERT Oracle Secure Coding Standard for Java (2011) Chapter 16 - Platform Security (SEC)
MemberOf	C	865	2011 Top 25 - Risky Resource Management
MemberOf	V	884	CWE Cross-section
MemberOf	C	991	SFP Secondary Cluster: Tainted Input to Environment
MemberOf	C	1354	OWASP Top Ten 2021 Category A08:2021 - Software and Data Integrity Failures
MemberOf	C	1364	ICS Communications: Zone Boundary Failures
MemberOf	C	1411	Comprehensive Categorization: Insufficient Verification of Data Authenticity

Publikusan ismert sérülékenységek

- Olyan (általában műszaki jellegű) sérülékenységek, melyeket nyilvánosságra hoztak, így bárki (beleértve a támadót) számára ismertek lehetnek
- Sérülékenységek rutinszerűen kerülnek így nyilvánosságra, általában egy kontrollált folyamaton (responsible disclosure procedure) keresztül
- A publikus sérülékenységek egyedi azonosítót kapnak
  - CVE ID – Common Vulnerabilities and Exposures (cve.mitre.org)
- A publikus sérülékenységekkel kapcsolatos információkat publikus sérülékenység-adatbázisokban tárolják
  - strukturált, kereshető
  - példa: US National Vulnerability Database (nvd.nist.gov)
- A nyilvánosságra hozatal lehetővé teszi az ismert sérülékenységekből származó problémák eliminálását, mely önmagában drasztikusan csökkenti a biztonsági kockázatot
  - Sajnos vannak olyan rendszerek, ahol az ismert sérülékenységek eliminálása lassú vagy nem is lehetséges, más megfontolások, szabályok miatt

CVE

- Common Vulnerabilities and Exposures
  - Nyilvánosan közzétett kiberbiztonsági sebezhetőségek azonosítása, meghatározása és katalogizálása
    - » sebezhetőségek egységes leírása
    - » egyedi azonosító biztosítja, hogy a szakemberek tudják, hogy ugyanazt a kérdést tárgyalják
  - Közösségi erőforrás, szabadon elérhető, kereshető, on-line: **https://cve.mitre.org/**



CVE – példa

CVE-ID	
CVE-2021-22909	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

A vulnerability found in EdgeMAX EdgeRouter V2.0.9 and earlier could allow a malicious actor to execute a man-in-the-middle (MitM) attack during a firmware update. This vulnerability is fixed in EdgeMAX EdgeRouter V2.0.9-hotfix.1 and later.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:https://community.ui.com/releas
- URL:https://community.ui.com/releases

Assigning CNA

HackerOne

Date Record Created

20210106	Disclaimer: The indicate when t

Posted 2 years ago

Security Advisory Bulletin 018

13 5k Comment Follow

Overview

First Published: May 18, 2021

Version: 1.0

Revision: 1.0

Summary

A vulnerability found in EdgeMAX EdgeRouter **V2.0.9 and earlier** could allow a malicious actor to execute a man-in-the-middle (MitM) attack during a firmware update. This vulnerability is fixed in **EdgeMAX EdgeRouter V2.0.9-hotfix.1 and later**.

Affected Products:

All EdgeMAX EdgeRouters

Mitigation:

Update the EdgeMax EdgeRouter to **V2.0.9-hotfix.1 or later**.

NIST NVD – példa

CVE-2021-22909 Detail

Description

A vulnerability found in EdgeMAX EdgeRouter V2.0.9 and earlier could allow a malicious actor to execute a man-in-the-middle (MitM) attack during a firmware update. This vulnerability is fixed in EdgeMAX EdgeRouter V2.0.9-hotfix.1 and later.

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 7.5 HIGH **Vector:** CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:H

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

QUICK INFO

CVE Dictionary Entry:
CVE-2021-22909
NVD Published Date:
05/27/2021
NVD Last Modified:
06/08/2021
Source:
HackerOne

NIST NVD – példa

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource
https://community.ui.com/releases/Security-Advisory-Bulletin-018-018/cfa1566b-4bf8-427b-8cc7-8cffba3a93a4	Vendor Advisory

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-295	Improper Certificate Validation	NIST
CWE-300	Channel Accessible by Non-Endpoint	HackerOne

Known Affected Software Configurations Switch to CPE 2.2

Configuration 1 (hide)

cpe:2.3:o:ui:edgemax_edgerouter_firmware:*:*:*:*:*:*:*:*	**Up to (including)**
Show Matching CPE(s)	**2.0.9**
Running on/with	
cpe:2.3:h:ui:edgemax_edgerouter:-:*:*:*:*:*:*:*	
Show Matching CPE(s)	

Zero-day sérülékenységek

- Olyan sérülékenységek, melyek csak a támadó számára ismertek
  - vannak olyan cégek, melyek abból élnek, hogy sérülékenységeket keresnek és adnak el potenciális támadóknak (pl. bűnözőknek, kormányzatoknak)
- A 0-day sérülékenységek nagy problémát jelentenek, mert nem lehet rájuk hatékonyan felkészülni, nem lehet ellenük védelmet kialakítani
- Szerencsére a 0-day sérülékenységek ritkák és drágák, ezért főként célzott támadásokban használják őket, ahol
  - a támadás sikere nagyon fontos
  - a támadás észlelésének (és így a sérülékenység napvilágra kerülésének) esélye viszont elég alacsony

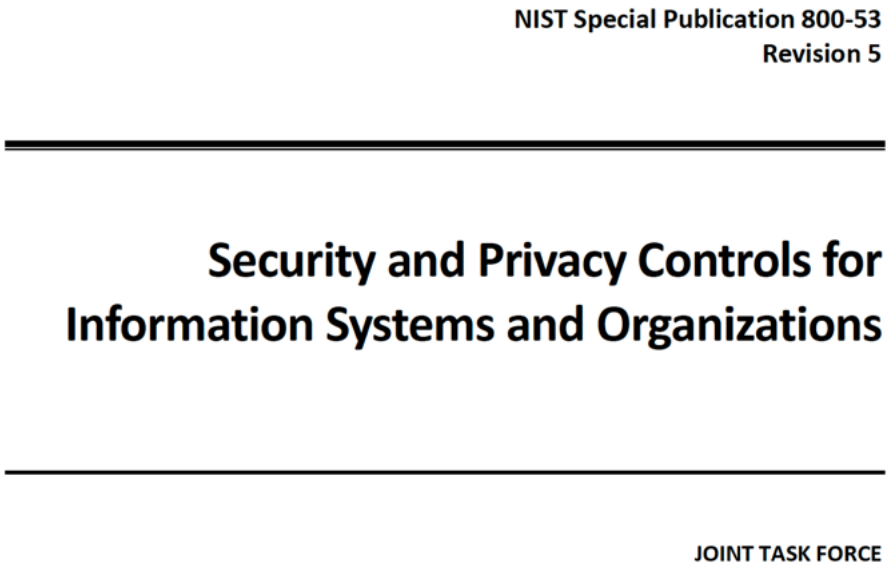
Ellenintézkedések

Az ellenintézkedések fajtái

- **Műszaki (logikai) –** a hosztokon és a hálózatokban használt biztonsági mechanizmusok, megoldások
  - pl. tűzfalak, anti-vírus szoftverek, jelszavas hitelesítés, kritpográfiai algoritmusok és protokollok, ...
- **Fizikai –** fizikai védelmet biztosító mechanizmusok
  - pl. zárak, kerítések, biztonsági őrök, bontásellenálló (tamper resistant) hardver elemek, ...
- **Üzemeltetési –** a rendszer üzemeltetésével és a személyzet menedzsmentjével kapcsolatos szabályzatok és eljárásrendek
  - pl. jelszócserére és –erősségre vonatkozó szabályozás, rendszeres biztonsági tesztelés, ...
  - pl. felvétel és elbocsátás során alkalmazott eljárások, kötelező szabadság, ...
- **Személyi –** a személyzet biztonságtudatosságának, megbízhatóságának, lojalitásának növelését célzó mechanizmusok
  - pl. tréning, gyakorlat, jó fizetés, vonzó munkakörülmények, ...

NIST Special Publication 800-53 (Rev 5)

- Információs rendszerek és szervezetek biztonsági és adatvédelmi mechanizmusainak (kontrollok) katalógusa
- A felsorolt mechanizmusok védelmet nyújtanak a különféle fenyegetések és kockázatok - köztük ellenséges támadások, emberi hibák, természeti katasztrófák - ellen
- A felsorolás kimerítő jellegű, a mechanizmusok a küldetésből és az üzleti igényekből, törvényekből, végrehajtási utasításokból, irányelvekből, szabályzatokból, szabványokból és iránymutatásokból származó különböző követelményekre válaszolnak
- 20 család, 300+ mechanizmus, 492 oldal (!)


NIST Special Publication 800-53
Revision 5

Security and Privacy Controls for Information Systems and Organizations

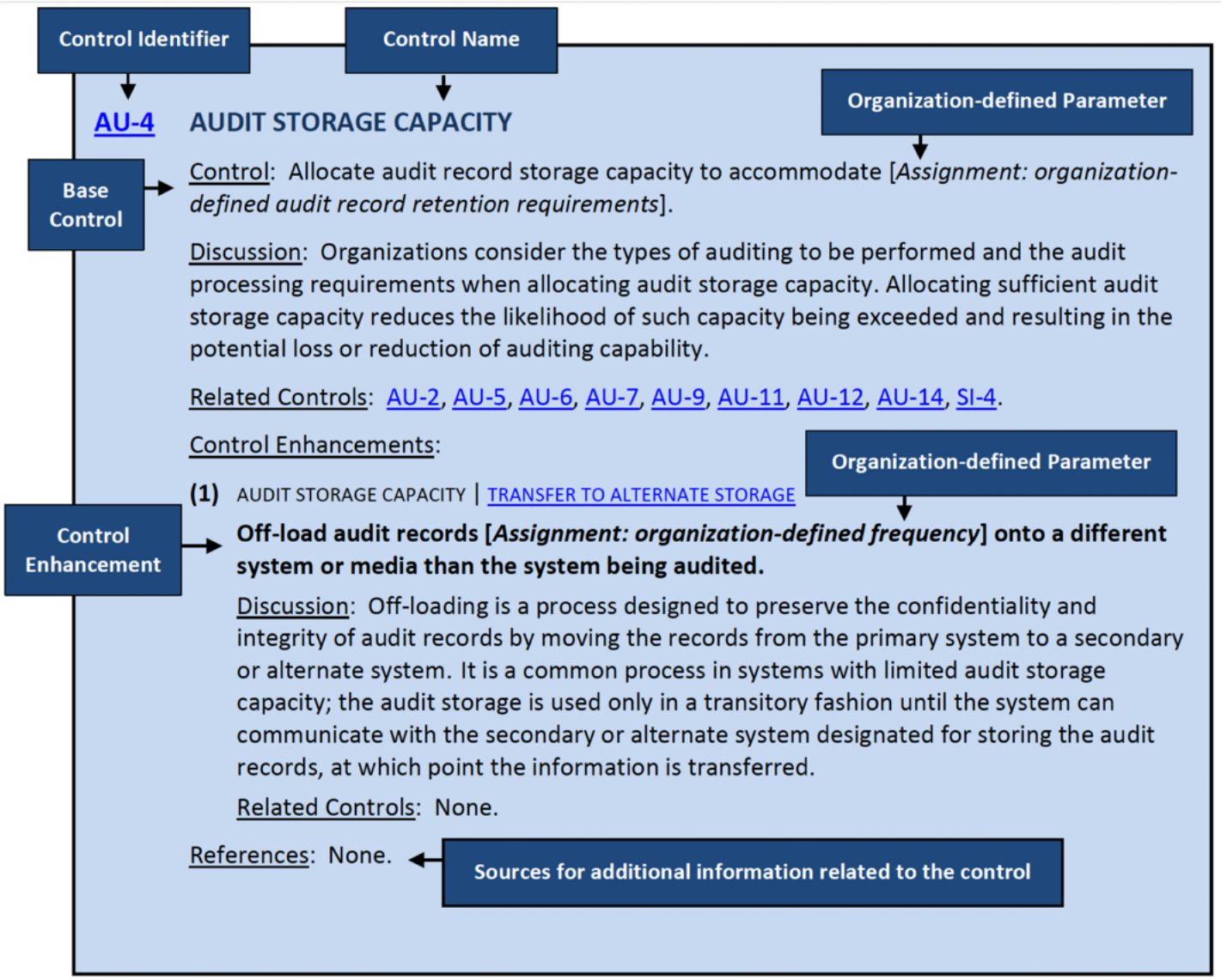
JOINT TASK FORCE

NIST SP800-53r5 – Mechanizmus családok

TABLE 1: SECURITY AND PRIVACY CONTROL FAMILIES

ID	FAMILY	ID	FAMILY
AC	Access Control	PE	Physical and Environmental Protection
AT	Awareness and Training	PL	Planning
AU	Audit and Accountability	PM	Program Management
CA	Assessment, Authorization, and Monitoring	PS	Personnel Security
CM	Configuration Management	PT	PII Processing and Transparency
CP	Contingency Planning	RA	Risk Assessment
IA	Identification and Authentication	SA	System and Services Acquisition
IR	Incident Response	SC	System and Communications Protection
MA	Maintenance	SI	System and Information Integrity
MP	Media Protection	SR	Supply Chain Risk Management

NIST SP800-53r5 – Egy leírás struktúrája



NIST SP800-53r5 – Mechanizmus leírás (példa)

SI-3 **MALICIOUS CODE PROTECTION**

Control:

a. Implement [*Selection (one or more): signature based; non-signature based*] malicious code protection mechanisms at system entry and exit points to detect and eradicate malicious code;

b. Automatically update malicious code protection mechanisms as new releases are available in accordance with organizational configuration management policy and procedures;

c. Configure malicious code protection mechanisms to:

   1. Perform periodic scans of the system [*Assignment: organization-defined frequency*] and real-time scans of files from external sources at [*Selection (one or more): endpoint; network entry and exit points*] as the files are downloaded, opened, or executed in accordance with organizational policy; and

   2. [*Selection (one or more): block malicious code; quarantine malicious code; take* [*Assignment: organization-defined action*]]; and send alert to [*Assignment: organization-defined personnel or roles*] in response to malicious code detection; and

d. Address the receipt of false positives during malicious code detection and eradication and the resulting potential impact on the availability of the system.

Discussion: System entry and exit points include firewalls, remote access servers, workstations, electronic mail servers, web servers, proxy servers, notebook computers, and mobile devices. Malicious code includes viruses, worms, Trojan horses, and spyware. Malicious code can also be encoded in various formats contained within compressed or hidden files or hidden in files using techniques such as steganography. Malicious code can be inserted into systems in a variety of ways, including by electronic mail, the world-wide web, and portable storage devices. Malicious code insertions occur through the exploitation of system vulnerabilities. A variety of technologies and methods exist to limit or eliminate the effects of malicious code.

Malicious code protection mechanisms include both signature- and nonsignature-based technologies. Nonsignature-based detection mechanisms include artificial intelligence techniques that use heuristics to detect, analyze, and describe the characteristics or behavior of malicious code and to provide controls against such code for which signatures do not yet exist or for which existing signatures may not be effective. Malicious code for which active signatures do not yet exist or may be ineffective includes polymorphic malicious code (i.e., code that changes signatures when it replicates). Nonsignature-based mechanisms also include reputation-based technologies. In addition to the above technologies, pervasive configuration management, comprehensive software integrity controls, and anti-exploitation software may be effective in preventing the execution of unauthorized code. Malicious code may be present in commercial off-the-shelf software as well as custom-built software and could include logic bombs, backdoors, and other types of attacks that could affect organizational mission and business functions.

Related Controls: AC-4, AC-19, CM-3, CM-8, IR-4, MA-3, MA-4, PL-9, RA-5, SC-7, SC-23, SC-26, SC-28, SC-44, SI-2, SI-4, SI-7, SI-8, SI-15.

Control Enhancements:

(1) MALICIOUS CODE PROTECTION | CENTRAL MANAGEMENT

[Withdrawn: Incorporated into PL-9.]

(2) MALICIOUS CODE PROTECTION | AUTOMATIC UPDATES

[Withdrawn: Incorporated into SI-3.]

(3) MALICIOUS CODE PROTECTION | NON-PRIVILEGED USERS

[Withdrawn: Incorporated into AC-6(10).]

(4) MALICIOUS CODE PROTECTION | UPDATES ONLY BY PRIVILEGED USERS

Update malicious code protection mechanisms only when directed by a privileged user.

Discussion: Protection mechanisms for malicious code are typically categorized as security-related software and, as such, are only updated by organizational personnel with appropriate access privileges.

Related Controls: CM-5.

(5) MALICIOUS CODE PROTECTION | PORTABLE STORAGE DEVICES

(6) MALICIOUS CODE PROTECTION | TESTING AND VERIFICATION

(a) Test malicious code protection mechanisms [*Assignment: organization-defined frequency*] by introducing known benign code into the system; and

(b) Verify that the detection of the code and the associated incident reporting occur.

Discussion: None.

Related Controls: CA-2, CA-7, RA-5.

(7) MALICIOUS CODE PROTECTION | NONSIGNATURE-BASED DETECTION

[Withdrawn: Incorporated into SI-3.]

(8) MALICIOUS CODE PROTECTION | DETECT UNAUTHORIZED COMMANDS

(a) Detect the following unauthorized operating system commands through the kernel application programming interface on [*Assignment: organization-defined system hardware components*]: [*Assignment: organization-defined unauthorized operating system commands*]; and

(b) [*Selection (one or more): issue a warning; audit the command execution; prevent the execution of the command*].

Discussion: Detecting unauthorized commands can be applied to critical interfaces other than kernel-based interfaces, including interfaces with virtual machines and privileged applications. Unauthorized operating system commands include commands for kernel functions from system processes that are not trusted to initiate such commands as well as commands for kernel functions that are suspicious even though commands of that type are reasonable for processes to initiate. Organizations can define the malicious commands to be

NIST SP800-53r5 – Kapcsolódó szabványok

- NIST SP 800-53B: Control Baselines for Information Systems and Organizations (October 2020)
  - Biztonsági és adatvédelmi alap-mechanizmusok gyűjteménye (baseline) (az USA szövetségi kormánya számára)
    - » 3 biztonsági baseline különböző típusú rendszerekhez
    - » 1 adatvédelmi baseline
  - A baseline a NIST SP 800-53-ban szereplő mechanizmusok olyan minimális részhalmaza, amelyet egy csoport, szervezet vagy érdekközösség védelmi igényeinek kielégítésére válogattak össze
  - A dokumentum útmutatást nyújt továbbá a testreszabáshoz, ezzel megkönnyítve a mechanizmusok kiválasztási folyamatát és a baseline testreszabását az egyes érdekközösségek és működési környezetek (pl. tárgyak internete, CPS) számára.

- NIST SP 800-53Ar5: Assessing Security and Privacy Controls in Information Systems and Organizations (January 2022)
  - A rendszerekben és szervezetekben alkalmazott biztonsági és adatvédelmi mechanizmusok értékeléséhez ad módszertant és definiál eszközöket
  - Lehetővé teszi, hogy meggyőződjünk arról, hogy a kiválasztott biztonsági és adatvédelmi kontrollok végrehajtásra kerültek-e, és megfelelnek-e a kitűzött céloknak

Összefoglalás

- Az IT biztonság a szándékos támadásokból származó kompromittálódások (rendszer, információ) megelőzésével, észlelésével, illetve kezelésével foglalkozik
- A rendszer üzemeltetője szempontjából a biztonság lényegében kockázat menedzsmentet jelent
- A kockázat a sikeres támadásból származó várható veszteség, amit az alábbi tényezők befolyásolnak:
  - a potenciális veszteség mértéke
  - a támadást kivitelező támadó jellemzői (motiváció, képességek, erőforrások)
  - a rendszer sérülékenységei, melyeket a támadó kihasználhat
  - az alkalmazott ellenintézkedések, melyek a támadó dolgát nehezítik
- A kockázat menedzsment célja a kockázat minimalizálása adott költségvetés mellett
- A kockázat menedzsment egy ciklikus folyamat:
  - vagyonelemek/értékek felmérése → kockázat-becslés → kockázat-csökkentés → monitorozás és felülvizsgálat
  - **A biztonság tehát egy kedvező (kellően védett) állapot folyamatos fenntartását célzó folyamat eredménye**

Összefoglalás

- A kockázat menedzsment során tipikusan vizsgálandó kérdések:
  - Milyen vagyonelemek/értékek (hardware, szoftver, adat, humán) vannak a rendszerben?
  - Kik lehetnek plauzibilis támadók?
  - Milyen ismert és potenciális sérülékenységek vannak a rendszerben?
  - Mekkora az esélye annak, hogy ezeket a sérülékenységeket a plauzibilis támadók sikeresen ki tudják használni?
  - Mekkora lehet a sikeres támadásokból származó veszteség?
  - Mik a legnagyobb kockázatú fenyegetések?
  - Milyen ellenintézkedésekkel lehet a kockázatot hatékonyan elfogadható szintre csökkenteni?

- Az eredmény mindig valamilyen kompromisszum:
  - biztonság ←→ szolgáltatások, feature-ök, használhatóság, hatékonyság, ár...
  - Mindig lesz pozitív maradvány-kockázat, a kérdés, hogy elfogadható-e számunkra annak mértéke
  - Fel kell készülnünk arra, hogy lehetnek sikeres támadások → a hatásos és hatékony incidenskezelés alapvető fontosságú

Biztonsági incidensek kezelése

Biztonsági incidens

- Biztonsági incidens alatt egy sikeres támadásból származó, ***észlelt*** rendszer-kompromittálódást értünk

  Példák:

  – a Facebook fiókunkat feltörék (hogyan észlelhetjük?)
  – a fájljainkat rejtjelezte egy zsaroló virus (hogyan észlelhetjük?)
  – a webszerverünk le van terhelve, nem tudja kiszolgálni a legitim kéréseket (hogyan észlelhetjük?)

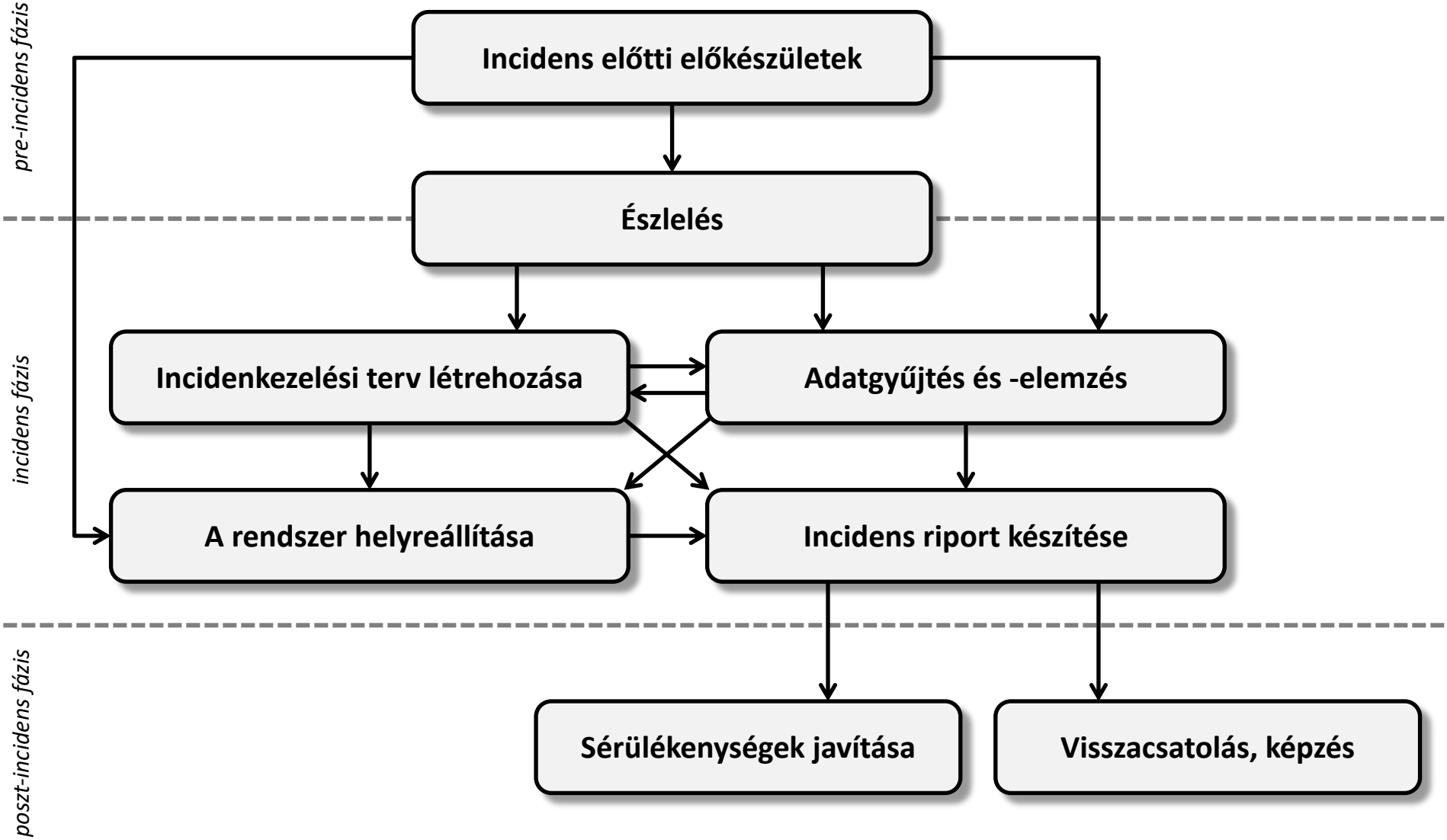
- A biztonsági incidenseknek komoly negatív hatása lehet az üzemeltetőre (cég, szervezet, egyén) nézve
  – közvetlen anyagi veszteség
  – jó hírnév elvesztése
  – személyes értékek (pl. pótolhatatlan személyes fotók) elvesztése

- Ezért a hatásos és hatékony incidenskezelés fontos feladat

Az incidenskezelés céljai

- Üzletmenet-folytonosság megszakadásának minimalizálása
  - A kompromittálódás behatárolása (containment)
  - A rendszer mielőbbi helyreállítása eredeti állapotába
- Hasonló incidensek jövőbeli bekövetkezésének megelőzése
  - Az incidens részleteinek megértése elemzés által
  - Az incidensben szerepet játszó sérülékenységek eliminálása
- Igazságügyi eljárás, nyomozás támogatása
  - Bizonyítékok gyűjtése, tárolása igazságügyi eljárásban felhasználható módon
  - Részletes incidens-elemzés és -riport

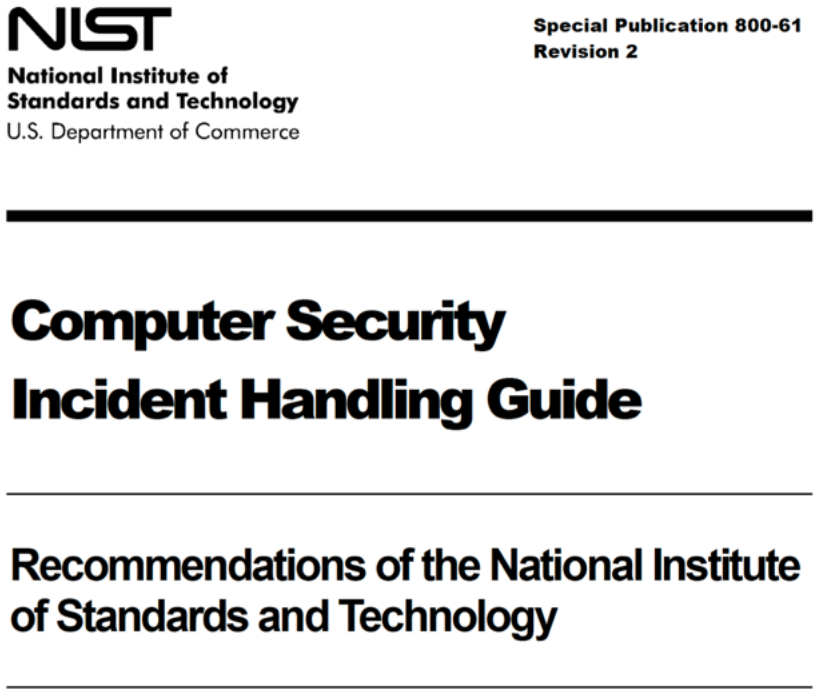
Az incidenskezelés lépései



bemenet vagy befolyás

NIST Special Publication 800-61 (Rev 2)

- Célja, hogy segítse a szervezeteket a számítógépes biztonsági incidensekre való reagálási képességek kialakításában és az incidensek hatékony és eredményes kezelésében
- Az incidenskezelési csoport (response team) felállításával, az incidenskezelési szabályzat és a kapcsolódó eljárások meghatározásával, valamint az incidensek kezelésére és az incidensekkel kapcsolatos információk megosztására vonatkozó hozzáállásokkal és módszerekkel foglalkozik



- Konkrét iránymutatásokat ad az incidensekkel kapcsolatos adatok elemzésére és az incidensekre adott megfelelő válaszlépések meghatározására

Ellenőrző kérdések

Alapfogalmak, kockázat menedzsment

- Milyen jellegű problémákkal foglalkozik az IT biztonság?
- Mi az a támadó, és miért lehetnek támadások sikeresek egy rendszer ellen?
- Hogyan védekezhet a rendszer üzemeltetője a támadások ellen?
- Mit értünk rendszer-kompromittálódás alatt?
- Definiálja az alábbi fontosabb fogalmakat:
  - Bizalmasság, integritás, rendelkezésre állás (CIA)
  - Hitelesítés, engedélyezés, felelőségre vonhatóság (AAA)
  - Security engineering, security operations
- Hogyan definiáljuk a biztonsági kockázatot?
- Mik a kockázatot befolyásoló tényezők?
- Mik a kockázat menedzsment főbb megválaszolandó kérdései?
- Mi az a maradvány-kockázat? Miért nem csökkenthető nullára?

Támadók

- Milyen tényezők alapján jellemezhetők a támadók?
- Milyen információkat érdemes megszerezni támadás előtt?
- A támadó műszaki képességeinek, szakértelmének milyen szintjei lehetnek?
- Miért fontos az anyagi erőforrás a támadó számára?
- Milyen tipikus támadó profilok vannak? Minden profil esetén foglalja össze az adott támadó típus motivációját, képességeit, és erőforrásait!
- Milyen szolgáltatások érhetők el a kiberbűnözők számára az underground piacon?
- Mik a célzott támadás jellemzői?
- Mi az a Stuxnet, és miért fontos?
- Mi az a „kill chain” modell?

Sérülékenységek, ellenintézkedések, incidensek

- Milyen típusú sérülékenységek lehetségesek IT rendszerekben?
- Mik a műszaki jellegű sérülékenységek okai?
- Hogyan kezeljük a publikusan ismert sérülékenységeket?
- Mik azok a 0-day sérülékenységek? Miért veszélyesek?
- Milyen kockázat csökkentő ellenintézkedések lehetségesek IT rendszerekben?
- Soroljon fel néhány megelőző jellegű és néhány észlelést célzó biztonsági mechanizmust!
- Mi az a biztonsági incidens?
- Miért fontos az incidensek hatásos és hatékony kezelése?
- Mik az incidenskezelés céljai?
- Mik az incidenskezelés főbb lépései?