



Anleitung 1

Elevation of Privilege Anleitung

1. Zeichnen Sie ein Diagramm des Systems, für das Sie ein Bedrohungsmodell erstellen möchten.
 2. Teilen Sie die Karten an 3 – 6 Spieler aus.
 3. Das Spiel beginnt mit der Karte "Tampering 3".
- Es wird im Uhrzeigersinn gespielt. Wer an der Reihe ist, versucht die Suite (z.B. Tampering) zu bedienen, mit der die Runde eingeleitet wurde. Wer nicht bedienen kann, spielt eine Karte einer anderen Suite. Wer mit der höchstwertigen Karte bedienen kann, gewinnt die Runde, außer eine "Elevation of Privilege" Karte wird gespielt, die dann gewinnt. Um eine Karte zu spielen, lesen Sie die Bedrohung auf der Karte laut vor. Können Sie die Bedrohung nicht auf Ihr System anwenden, geht das Spiel weiter. Wer die Runde gewinnt, wählt Karte (und Suite), mit der die nächste Runde begonnen wird. Machen Sie nach jeder Runde eine kurze Pause, um über die Bedrohungen nachzudenken.

Punkte:

- 1 wenn Sie die Bedrohung auf Ihr System anwenden können,
+1 für den Rundengewinn

Anleitung



Anleitung 2

Elevation of Privilege Varianten

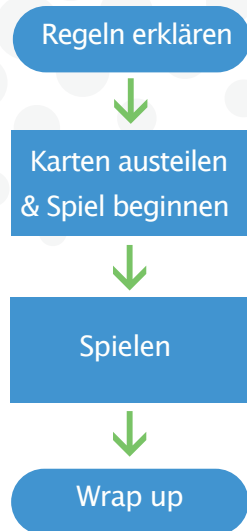
Optionale Varianten:

- a) Sie können Karten, die Sie nicht auf das System anwenden können, nach der 3. Runde abgeben. Vielleicht kann es jemand anders .
- b) Verdoppeln Sie die Punktezahlen und geben Sie 1 Punkt für Bedrohungen auf Karten anderer Spieler.
- c) Spieler die nicht an der Reihe sind können nach Spielen einer Karte binnen 1 Minute Varianten einer Bedrohung einwerfen und erläutern, die auf das System anwendbar sind. Geben Sie hierfür 1 Extrapunkt. Markieren Sie im Diagramm, wo die Bedrohung statt findet.

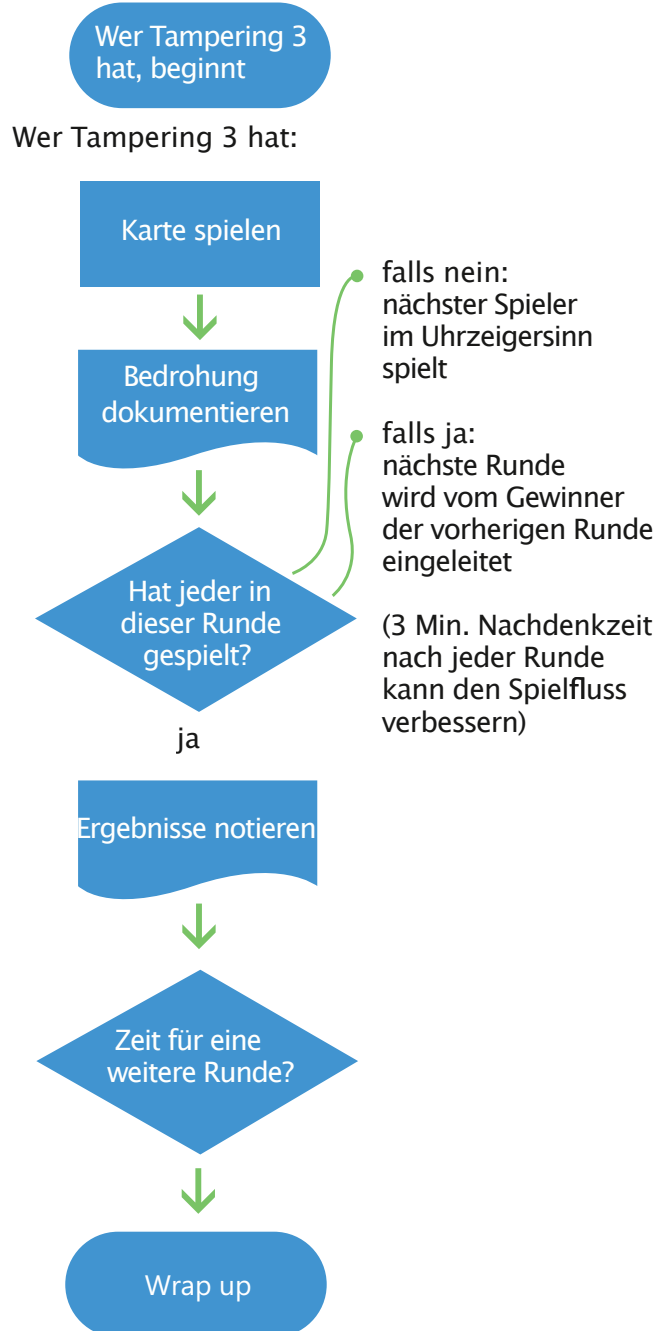
Thanks to Laurie Williams for inspiration.

Anleitung

Context



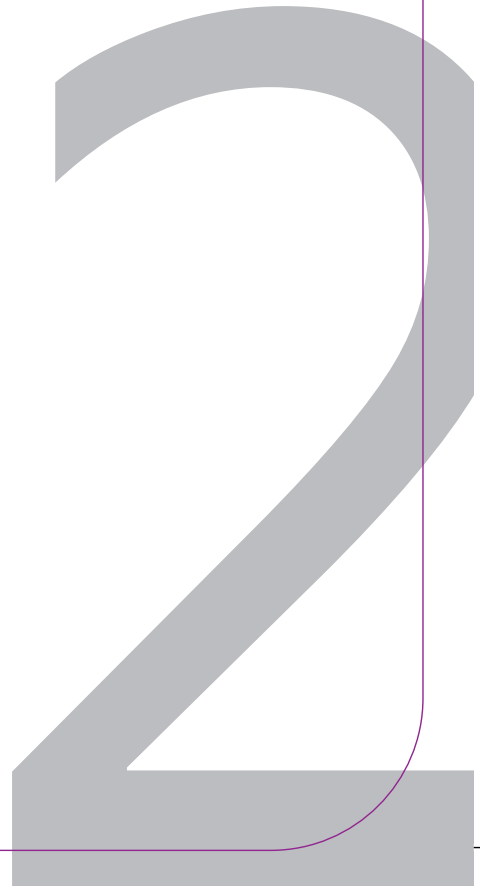
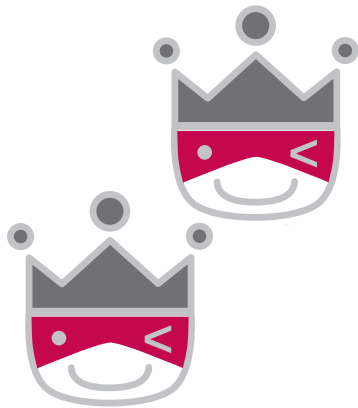
Play



2

Spoofing

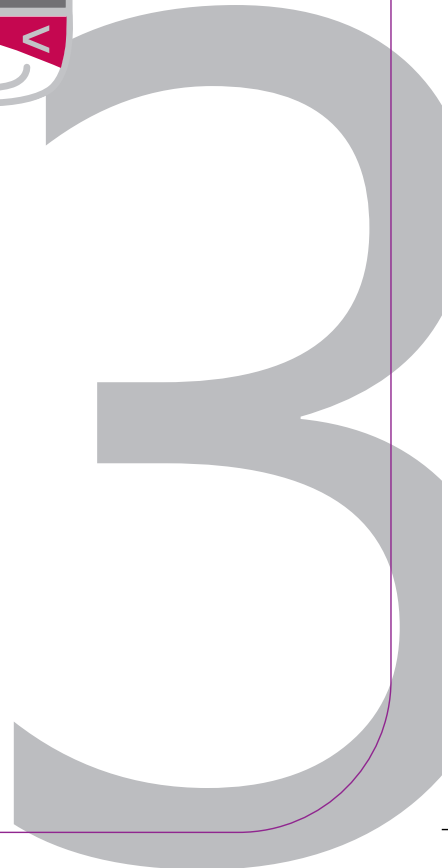
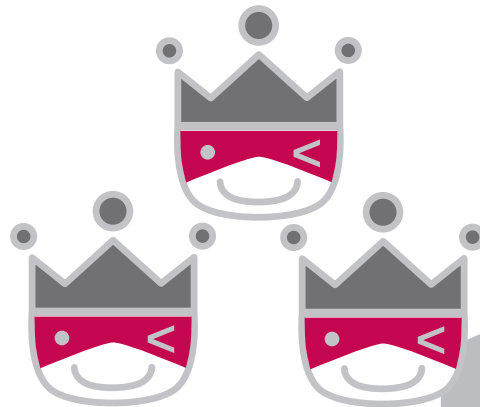
Ein Angreifer "sitzt" (lauscht) auf einem zufälligen Port oder Socket, den der Server üblicherweise nutzte.



3

Spoofing

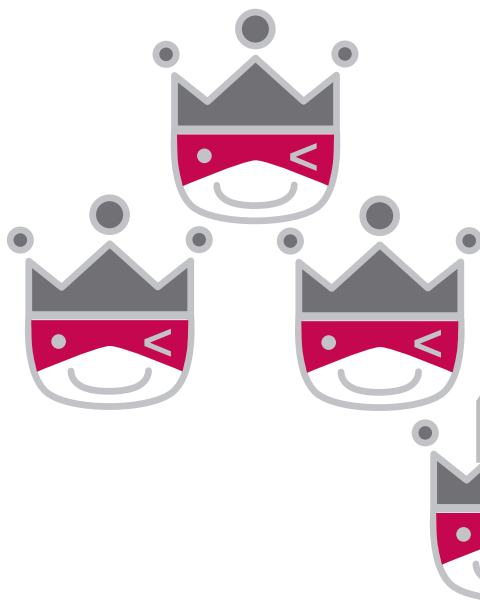
Ein Angreifer kann alle mögl. Credentials der Reihe nach durchprobieren und es gibt keinen Mechanismus, der ihn ausbremst (online oder offline).



4

Spoofing

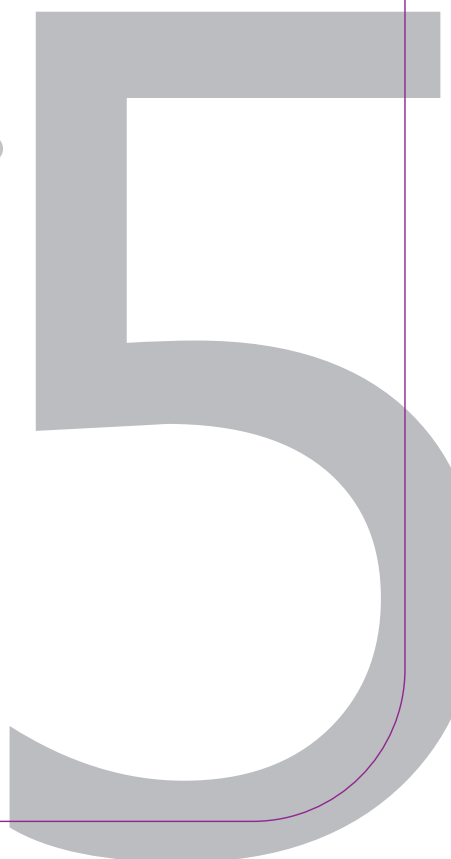
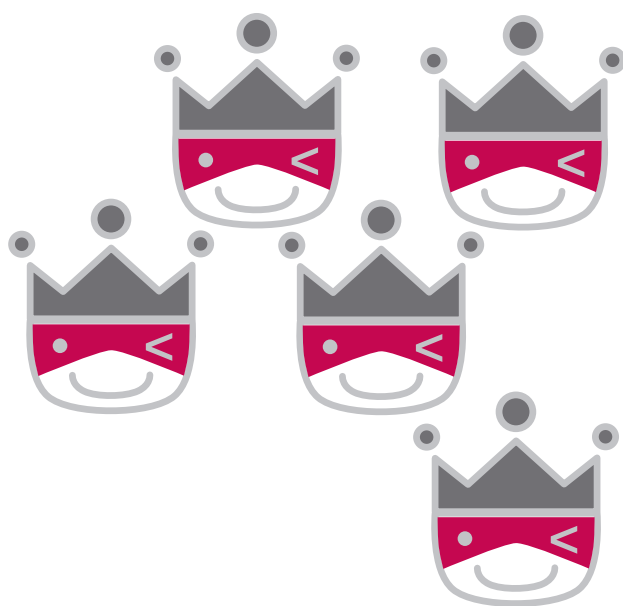
Ein Angreifer kann sich anonym verbinden, weil wir davon ausgehen, dass Authentisierung auf einer höheren Schicht stattfindet.



5

Spoofing

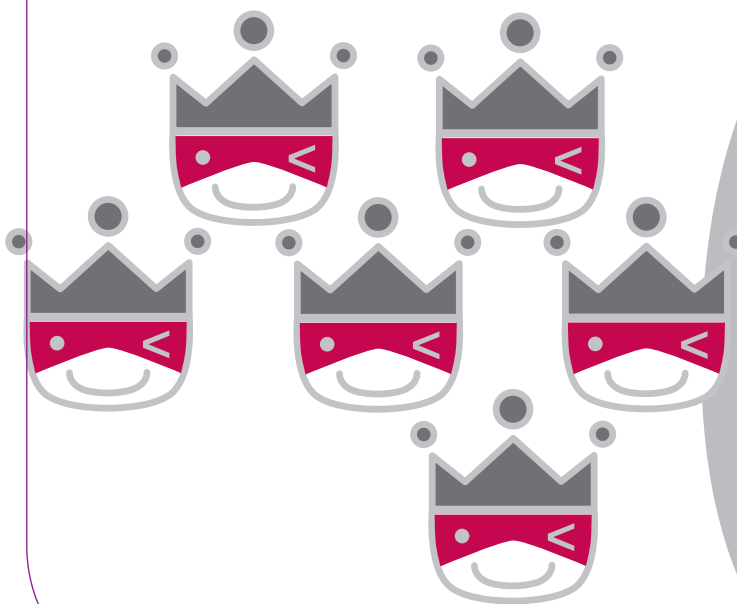
Ein Angreifer kann einen Client verwirren, weil es zu viele Wege gibt, einen Server zu identifizieren.



6

Spoofing

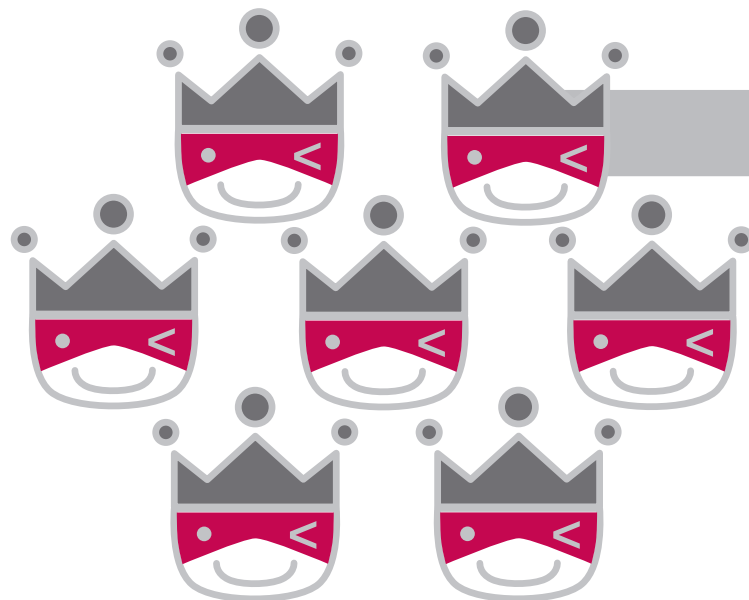
Ein Angreifer kann einen Server spoofen, weil auf dem Client keinerlei Identifizierungsmerkmale gespeichert sind, die bei erneuter Verbindung überprüft würden (es gibt keine Key persistence).



7

Spoofing

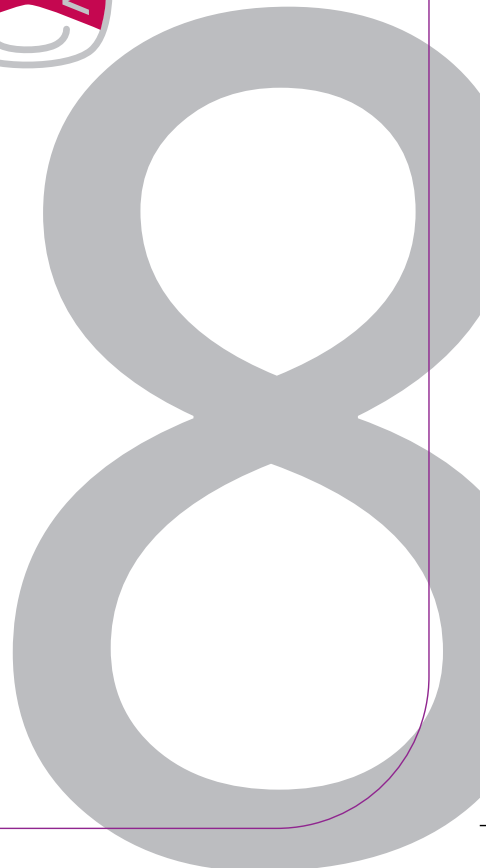
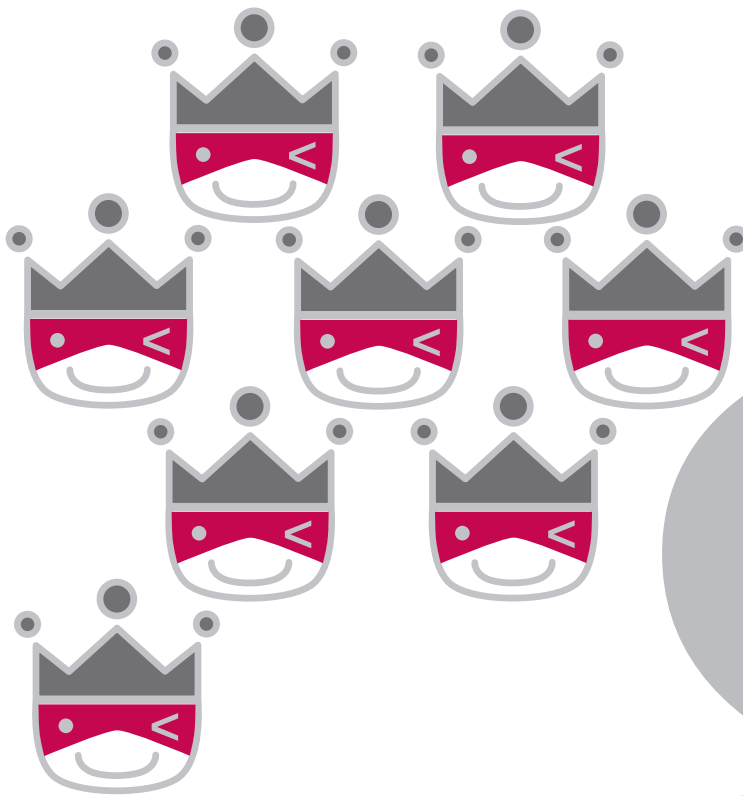
Ein Angreifer kann sich zu einem Server oder Peer über einen nicht authentisierten unverschlüsselten Kanal verbinden.



8

Spoofing

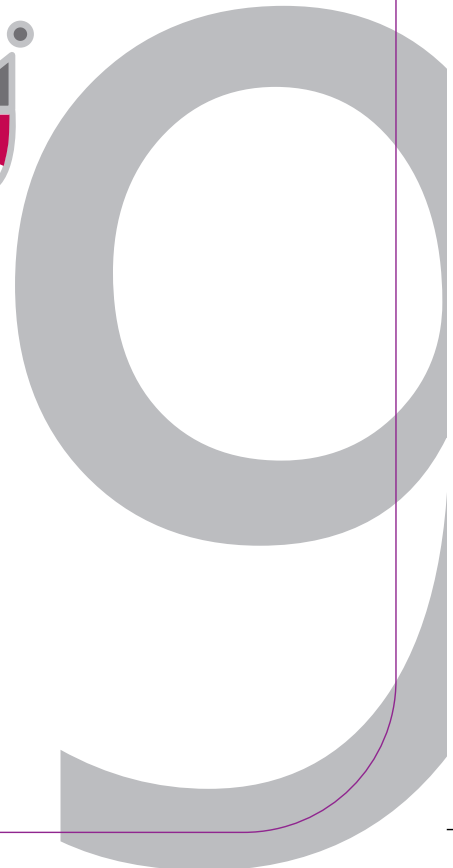
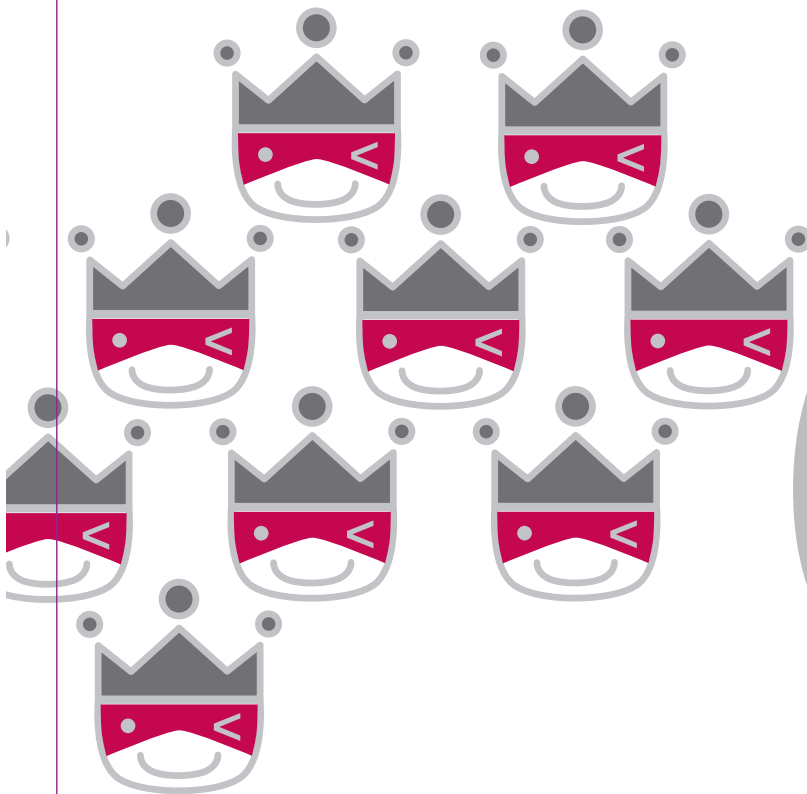
Ein Angreifer kann auf einem Server gespeicherte Credentials stehlen und wieder verwenden (z.B. Schlüssel in einer für andere lesbaren Datei).



9

Spoofing

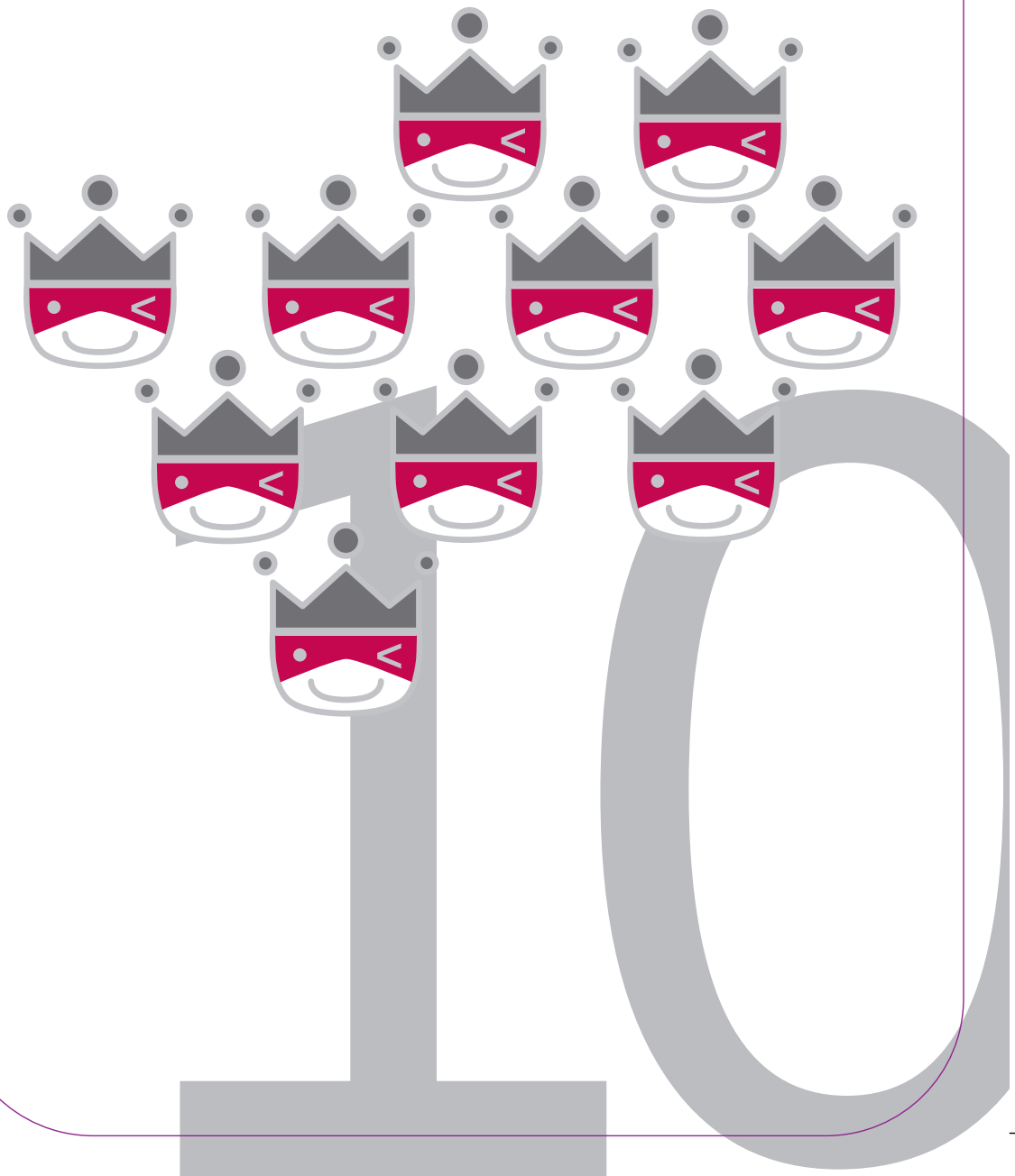
Ein Angreifer, der Zugang zu einem Passwort bekommt, kann es wieder verwenden (nutzen Sie stärkere Authentisierungsmethoden).



10

Spoofing

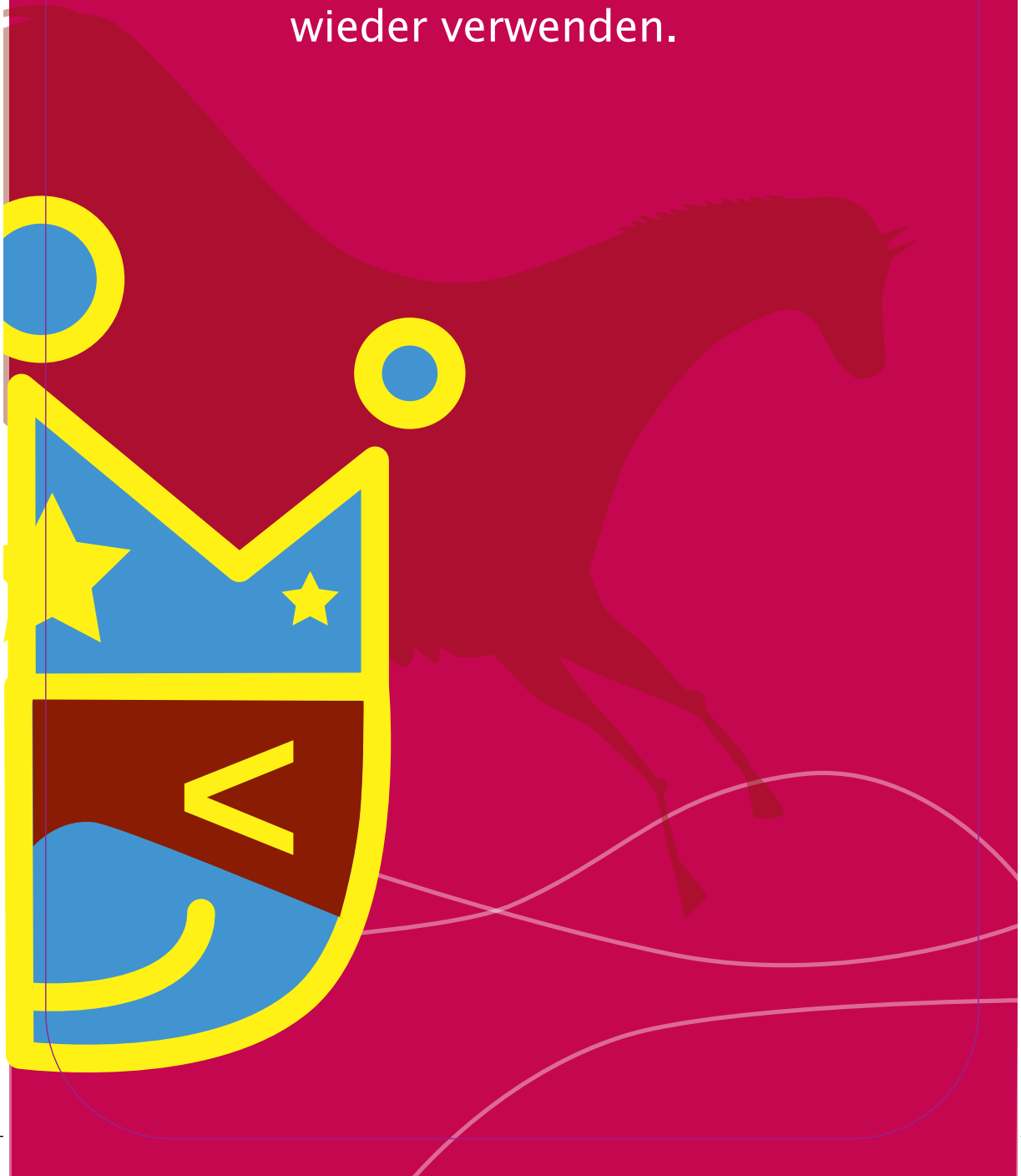
Ein Angreifer kann wählen, dass eine schwächere oder gar keine Authentisierung genutzt wird.





Spoofing

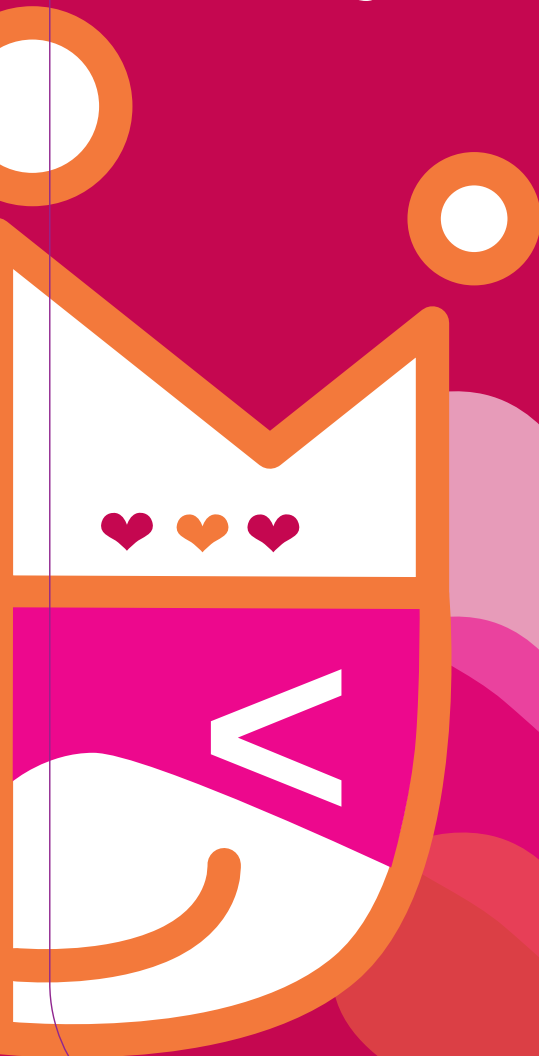
Ein Angreifer kann die auf einem Client gespeicherten Credentials stehlen und wieder verwenden.



Q

Spoofing

Ein Angreifer kann den Mechanismus angreifen, mit dem Passwörter zurückgesetzt oder aktualisiert werden (Account Recovery erfordert nicht die Eingabe des alten Passworts).

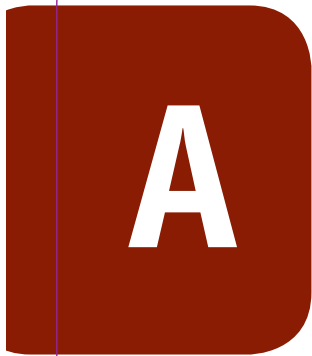




K

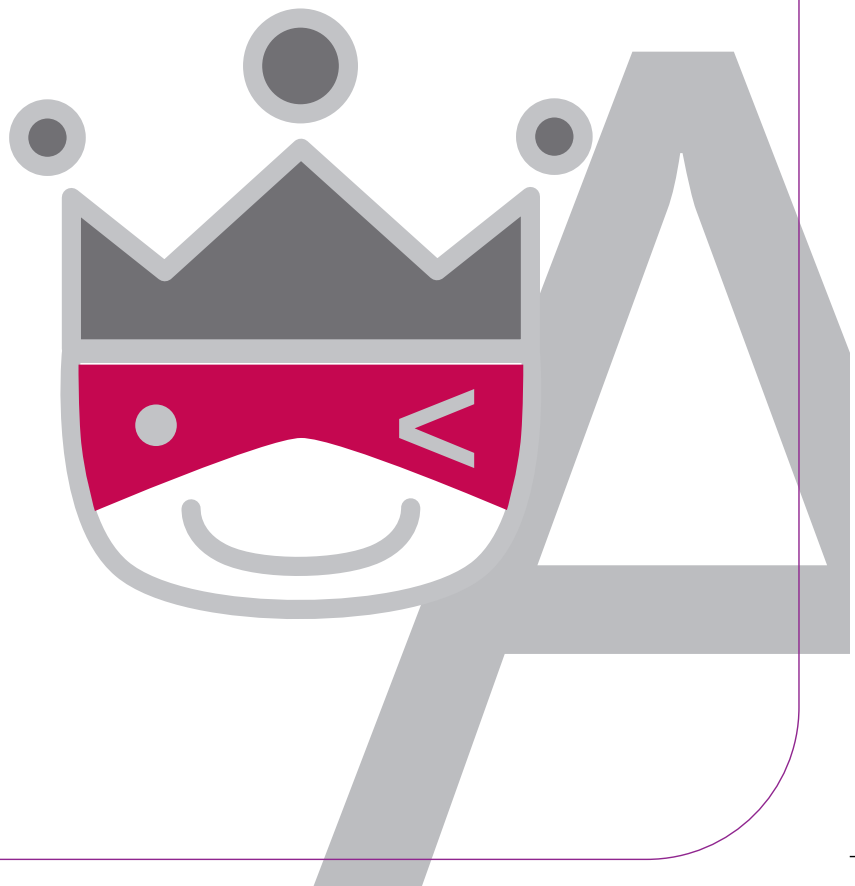
Spoofing

Ihr System wird mit einem Default Adminpasswort ausgeliefert und erzwingt nicht die Änderung dieses Passworts.



Spoofing

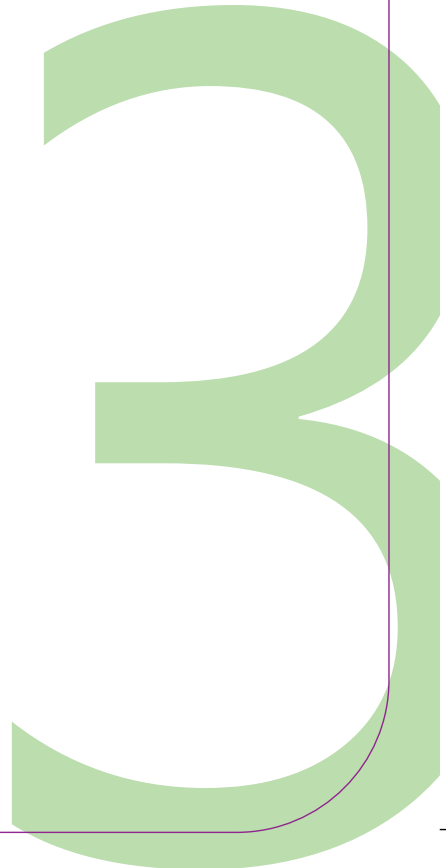
Sie haben einen neuen
Spoofing Angriff erfunden.



3

Tampering

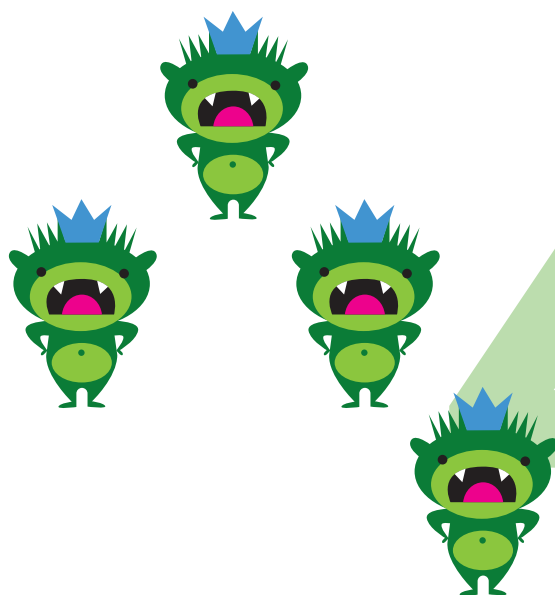
Statt auf Standard-Kryptografie zurück zu greifen, haben Sie sich selbst einen Mechanismus zur Gewährleistung von Integrität oder für den Schlüsselaustausch ausgedacht. Ein Angreifer kann sich diesen Umstand zunutze machen.



4

Tampering

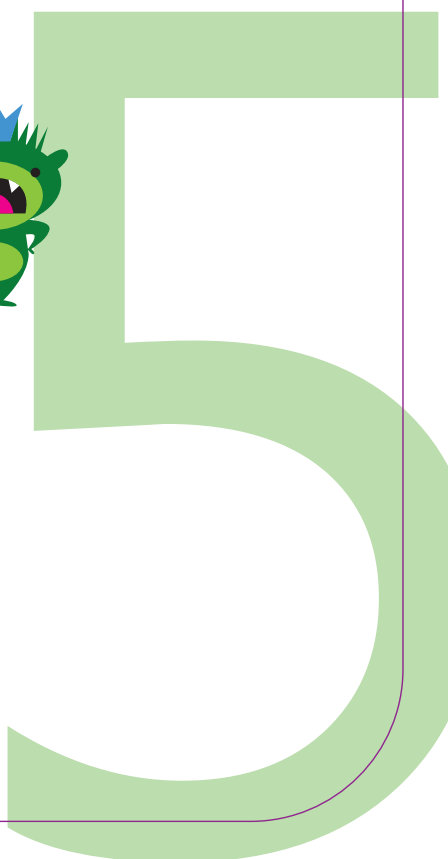
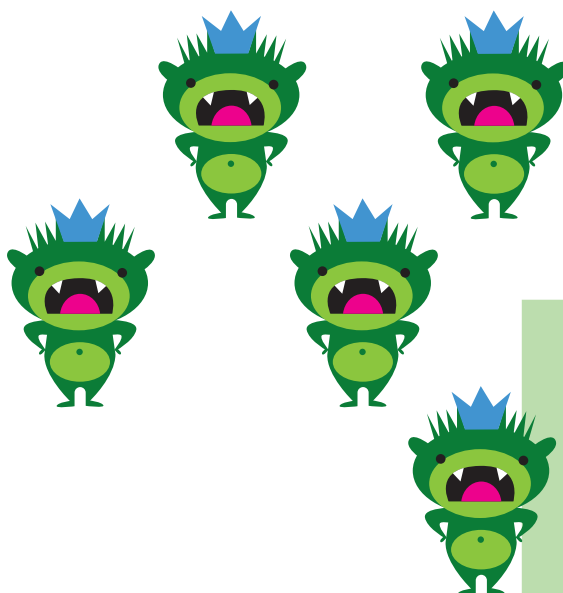
Ihr Code trifft Entscheidungen zur Zugangskontrolle an vielen unterschiedlichen Stellen, anstatt diese Funktion an zentraler Stelle (in einem Security Kernel) zu implementieren.



5

Tampering

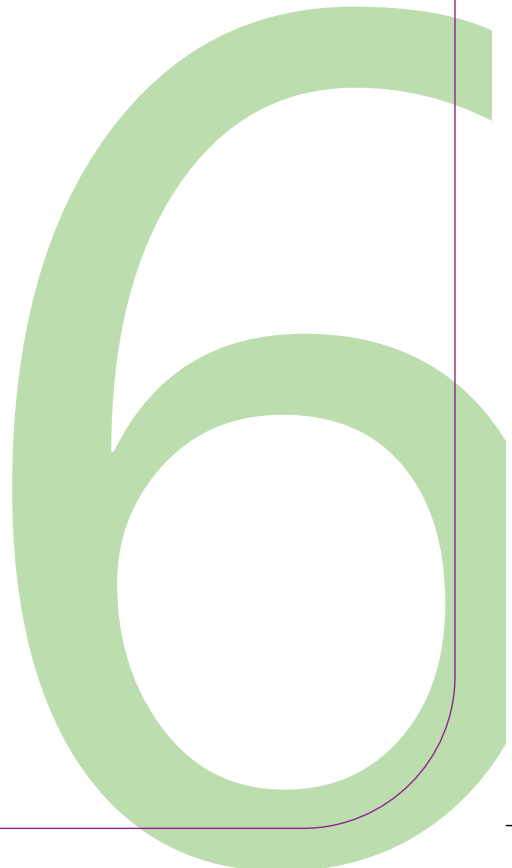
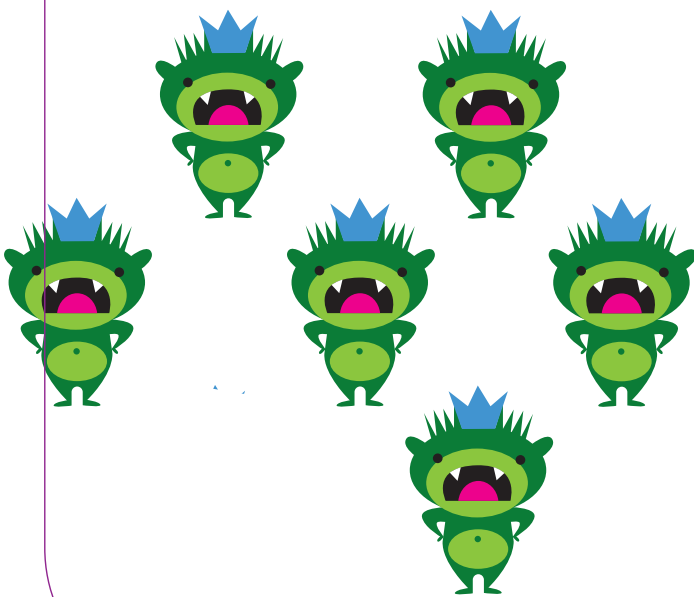
Ein Angreifer kann unbemerkt bereits übermittelte Daten erneut übertragen, weil Ihr Code keine Zeitstempel, Sequenznummern oder ähnliches nutzt, um dies zu verhindern oder zu erkennen.



6

Tampering

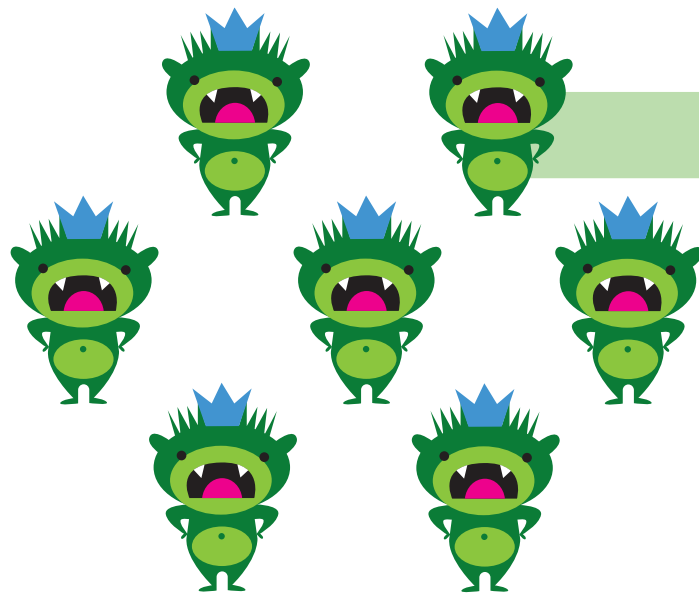
Ein Angreifer kann Daten an Speicherorten schreiben, an denen Ihr Code liegt oder die durch Ihren Code interpretiert werden.



7

Tampering

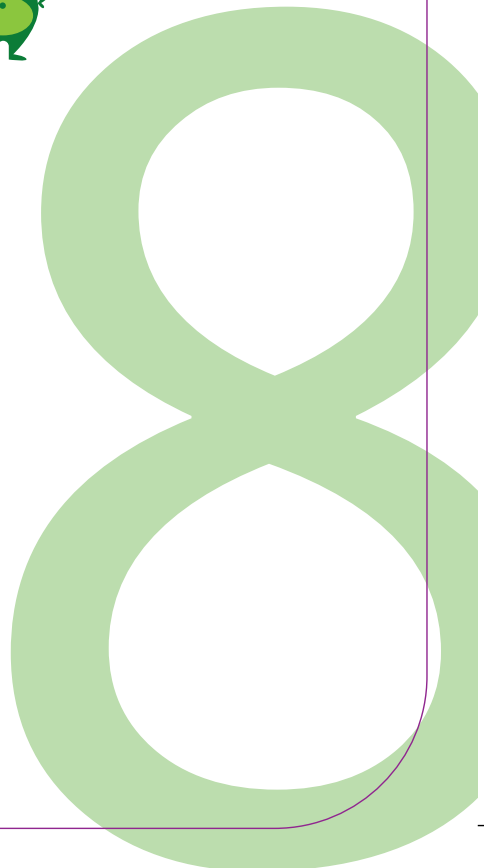
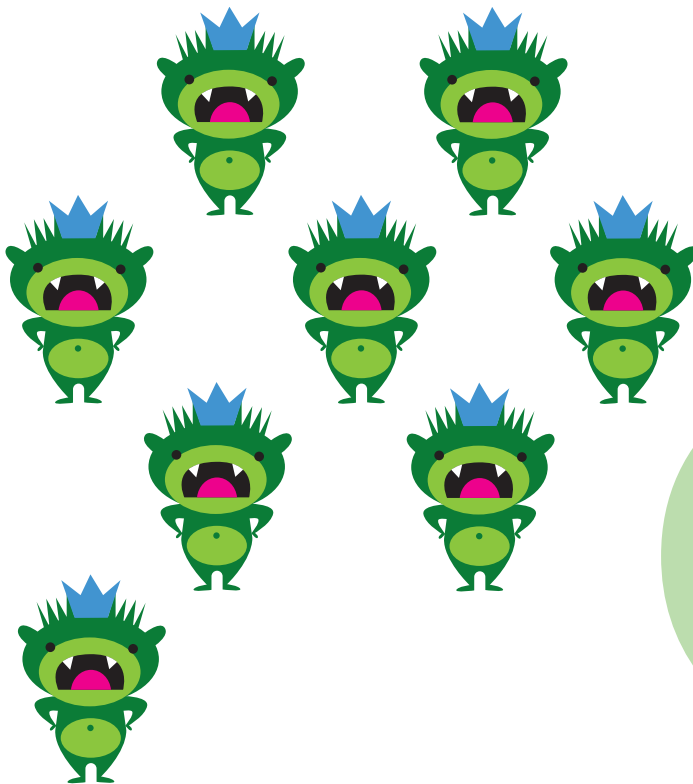
Ein Angreifer kann Berechtigungen umgehen, weil Sie Namen nicht kanonisieren (normalisieren), bevor Zugriffsrechte geprüft werden.



8

Tampering

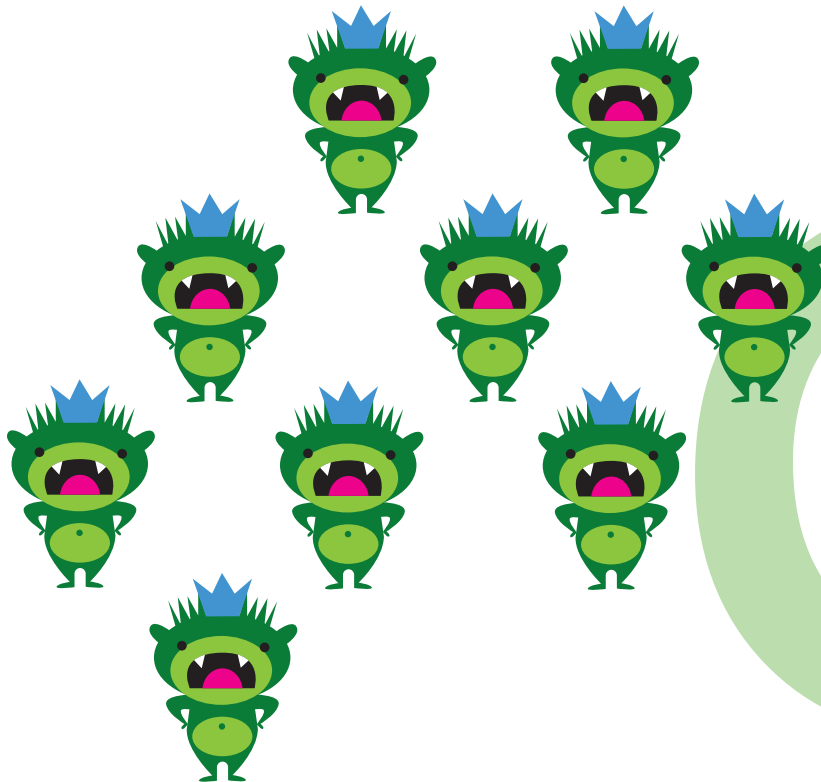
Ein Angreifer kann Daten manipulieren, die per Netzwerk übertragen werden, weil Ihr Code keine Integritätssicherung vorsieht.



9

Tampering

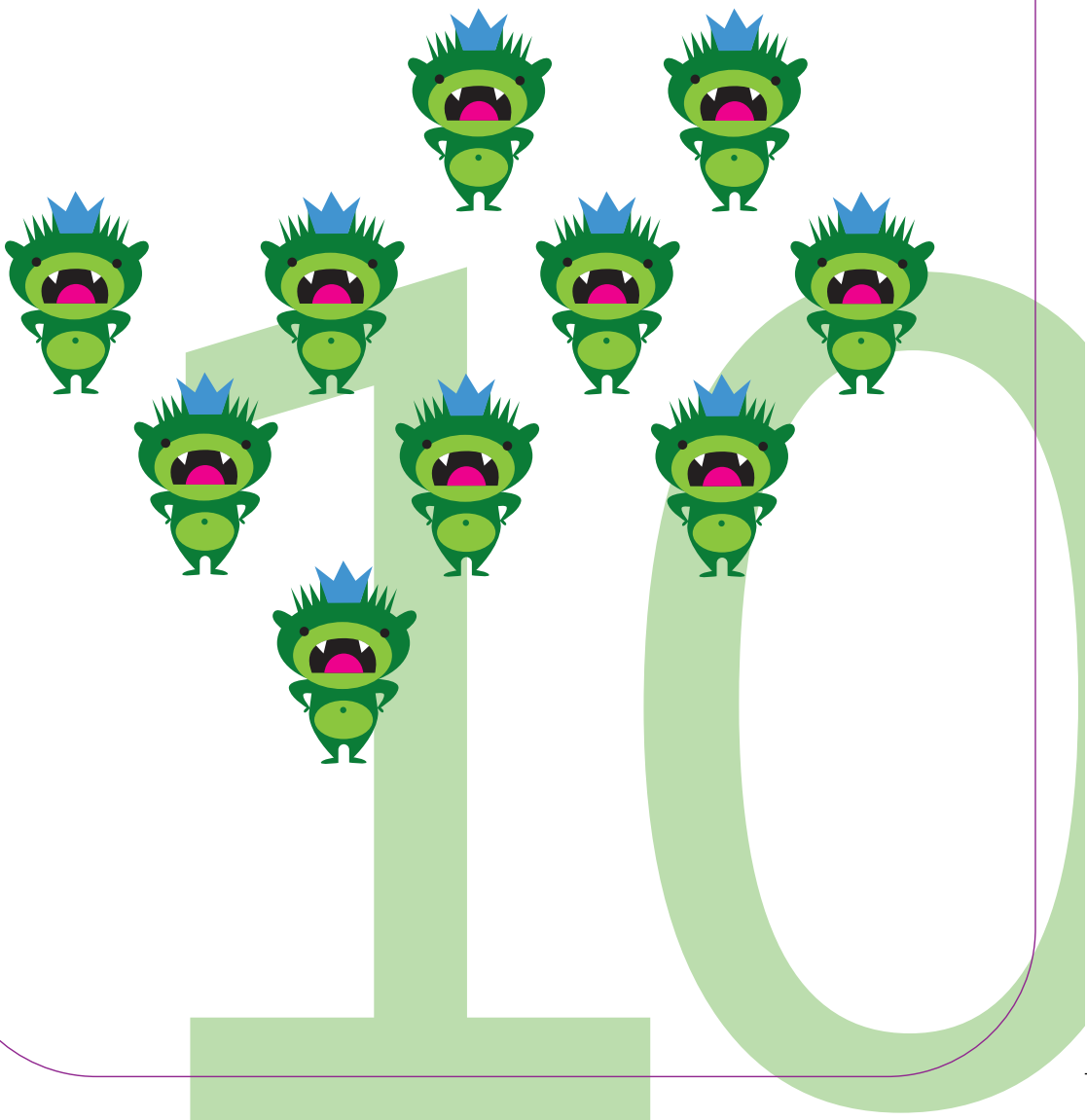
Ein Angreifer kann Status-
informationen beeinflussen.



10

Tampering

Ein Angreifer kann gespeicherte Daten verändern, weil die Berechtigungen (ACLs) zu wenig restriktiv sind oder eine Gruppe verwendet wird, die letztlich jedem Nutzer Zugriff gewährt.



J

Tampering

Ein Angreifer kann auf eine Ressource schreiben, weil es keine ACLs gibt, oder weil jeder berechtigt ist (world writable).



Q

Tampering

Ein Angreifer kann Parameter über eine Trust Boundary hinweg ändern, nachdem sie validiert wurden (z.B. in einem hidden field in HTML, oder einem Pointer an eine kritische Speicherstelle im RAM übergeben).



K

Tampering

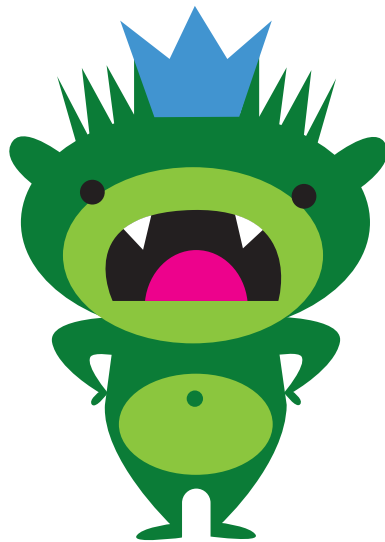
Ein Angreifer kann Code mithilfe eines Extension Points einbinden.



A

Tampering

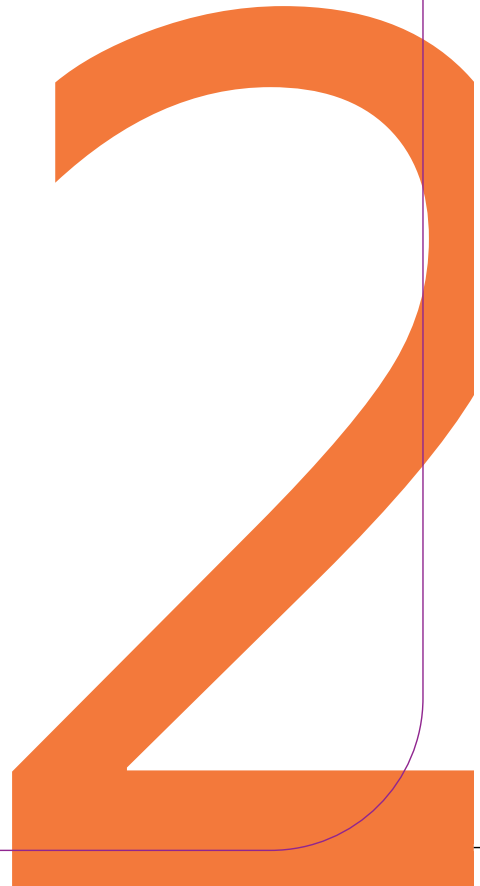
Sie haben einen neuen
Tampering Angriff erfunden.



2

Repudiation

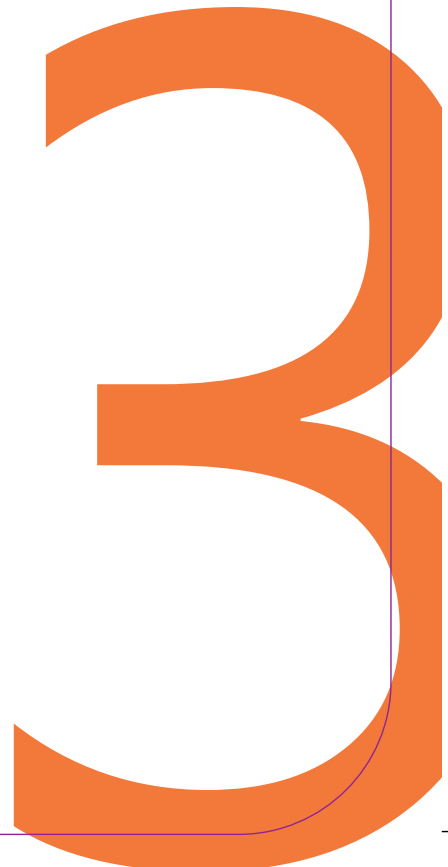
Ein Angreifer kann den Inhalt von Logdaten beeinflussen, einen Log Reader (Programm oder Nutzer) darüber angreifen und es ist nicht dokumentiert, ob und wie verschiedene Logdaten validiert werden.



3

Repudiation

Ein unprivilegiertes Nutzer oder Angreifer hat lesend Zugang zu interessanten Sicherheitsinformationen in den Logs.



4

Repudiation

Ein Angreifer kann digitale Signaturen manipulieren, weil Sie einen MAC Algorithmus statt einem Signierverfahren nutzen, oder weil Sie ein unsicheres Signierverfahren nutzen.



5

Repudiation

Ein Angreifer kann Lognachrichten verändern, die übers Netz übertragen werden, weil kein starker Mechanismus zur Gewährleistung der Integrität implementiert ist.



6

Repudiation

Ein Angreifer kann einen Logeintrag ohne Zeitstempel erzeugen (oder die Logs haben generell keine Zeitstempel).



7

Repudiation

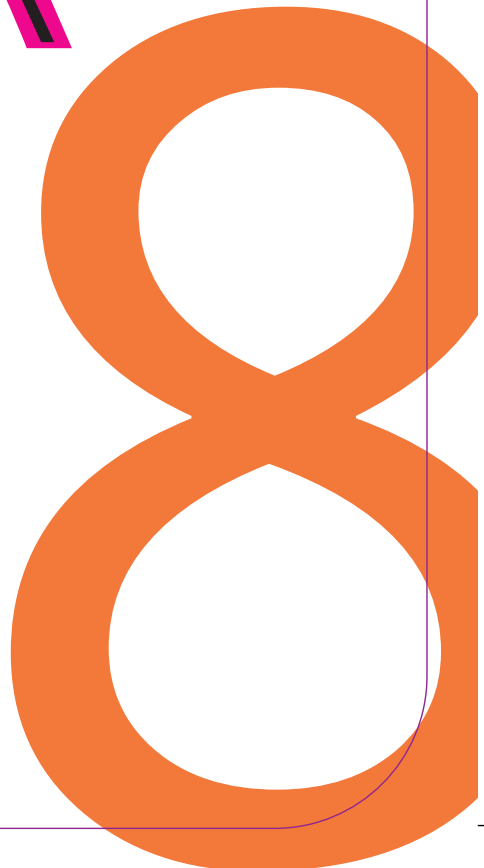
Ein Angreifer kann das Log zum Überlaufen bringen, so dass alte Logdaten überschrieben werden und somit verloren sind (wrap-around).



8

Repudiation

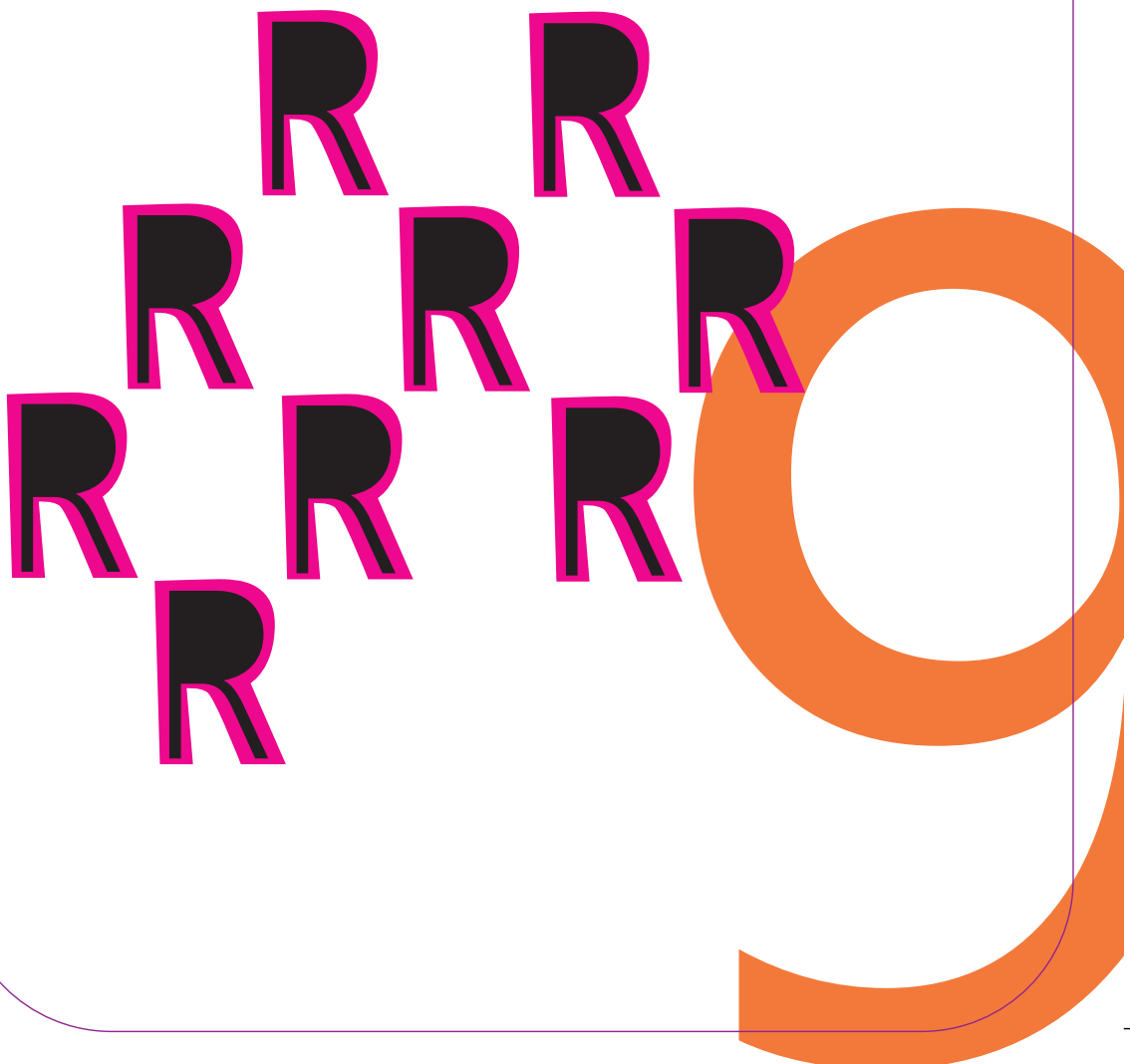
Ein Angreifer kann das Logging so austricksen, dass sicherheitsrelevante Logdaten nicht geschrieben werden oder durcheinander geraten.



9

Repudiation

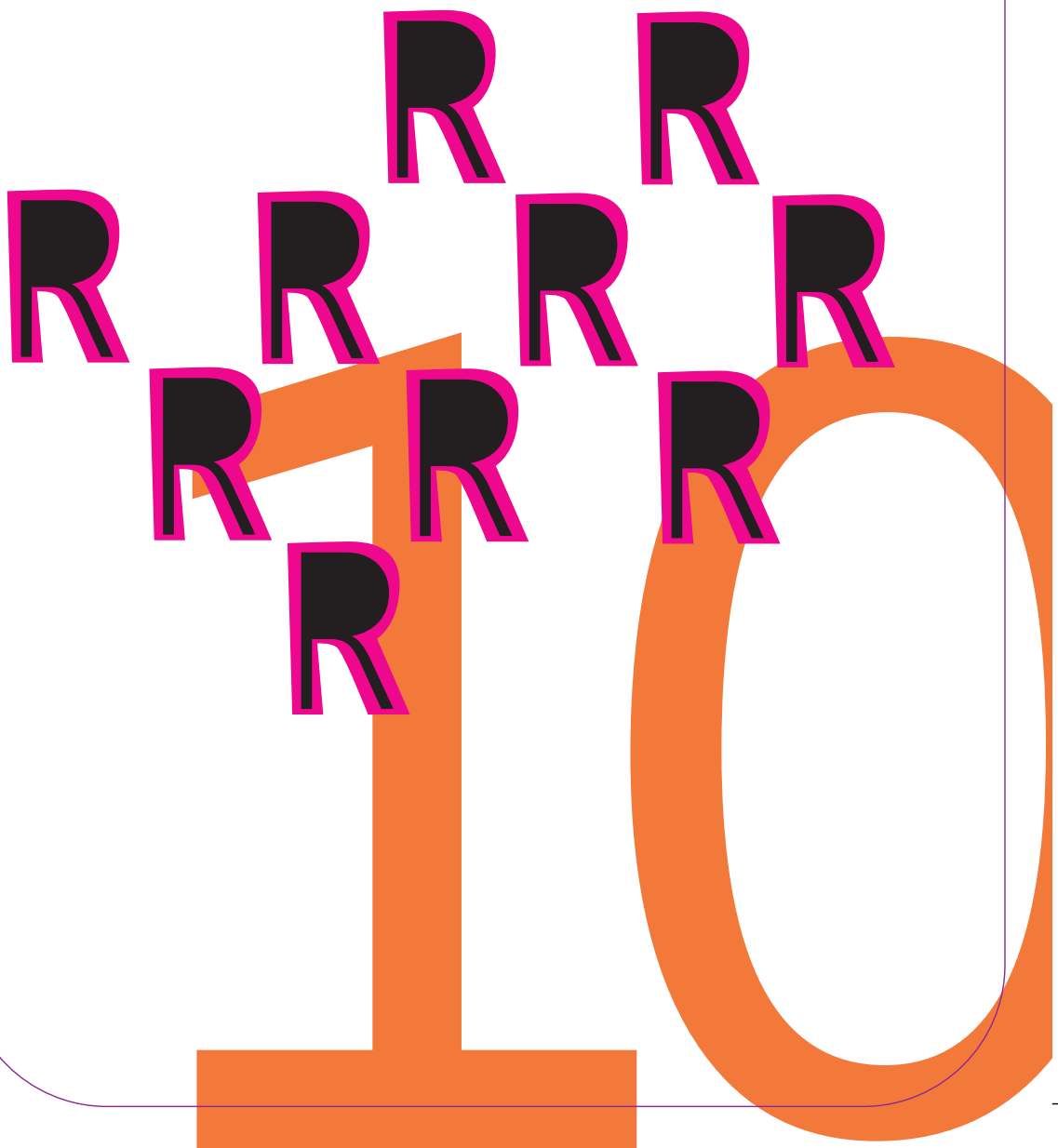
Ein Angreifer kann einen Shared key nutzen um sich als jemand anders auszugeben, so dass seine Aktionen ebenfalls unter dieser Identität mitgeloggt werden.



10

Repudiation

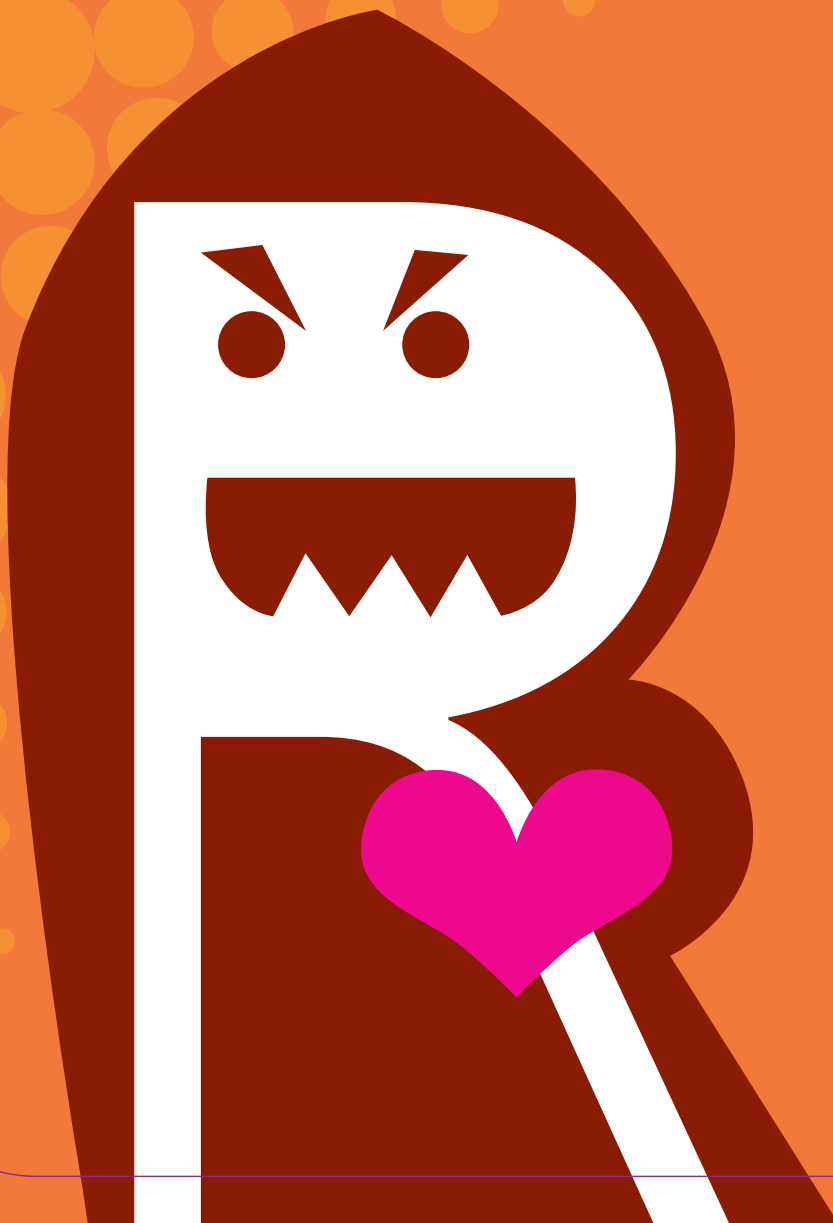
Ein Angreifer kann beliebige Logdaten in ein Logsystem einschleusen, weil die Logquellen nicht oder nur schwach authentisiert werden.



J

Repudiation

Ein Angreifer kann unbemerkt
Logs editieren oder löschen.





Q

Repudiation

Ein Angreifer kann abstreiten, etwas getan zu haben und es gibt keine brauchbaren Daten, um das Gegenteil zu beweisen.



**I didn't
do that.**



K

Repudiation

Das System hat keine Logs.

logs = 0



A

Repudiation

Sie haben einen neuen
Repudiation Angriff erfunden.

RA

2

Information Disclosure

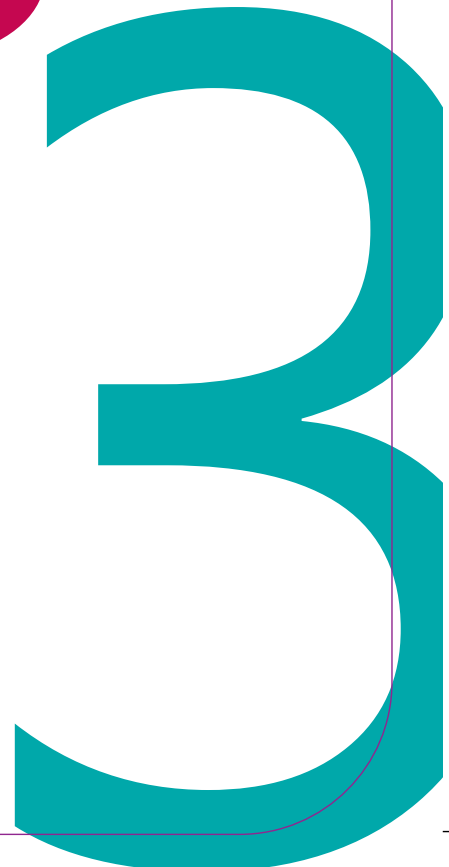
Ein Angreifer kann verschlüsselte Dateien mittels Brute-Force entschlüsseln, weil keine geeigneten Sicherheitsmaßnahmen dagegen vorhanden sind.



3

Information Disclosure

Ein Angreifer kann sicherheitsrelevante Fehlermeldungen sehen.



4

Information Disclosure

Ein Angreifer kann Dateninhalte lesen, weil die Nachrichten (z.B. E-Mails oder Cookies) nicht verschlüsselt sind, selbst wenn der Transportkanal verschlüsselt ist.



5

Information Disclosure

Ein Angreifer kann unter Umständen Daten lesen, die mit einem nicht standardisierten kryptografischem Algorithmus verschlüsselt sind.



6

Information Disclosure

Ein Angreifer kann Daten lesen, die lediglich versteckt oder verschleiert sind (z.B. für eine Undo-Funktion), so dass dem Nutzer gar nicht bewusst ist oder er vergessen hat, dass die Daten (noch) existieren.



7

Information Disclosure

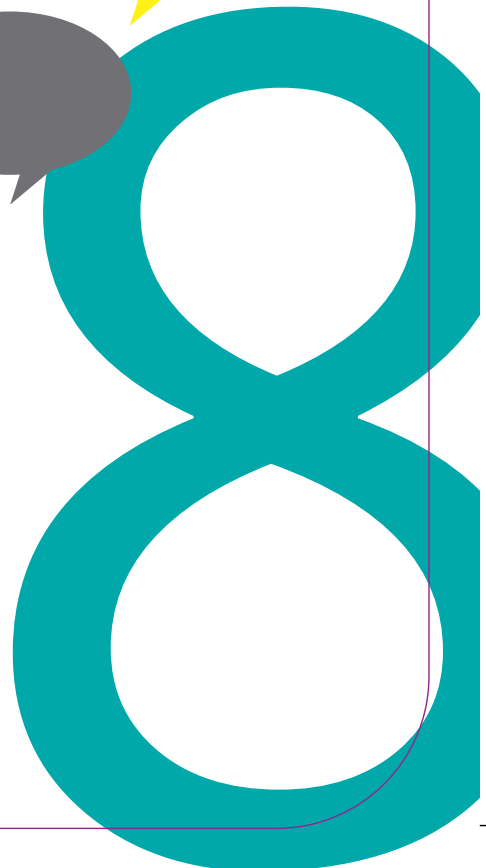
Ein Angreifer kann als "Man in the Middle" verschlüsselte Daten lesen, weil die Endpunkte einer Netzwerkverbindung nicht authentisiert sind.



8

Information Disclosure

Ein Angreifer kann (sensible) Informationen mithilfe eines Such-Indexers, Loggers oder eines anderen Mechanismus zugreifen.



9

Information Disclosure

Ein Angreifer kann sensible Informationen in einer Datei lesen, weil deren Zugriffsrechte falsch gesetzt sind (schwache ACL).



10

Information Disclosure

Ein Angreifer kann mangels Zugriffsbeschränkung eine sensible Datei lesen.



J

Information Disclosure

Ein Angreifer kann den statischen Schlüssel finden, der zur Verschlüsselung genutzt wird.

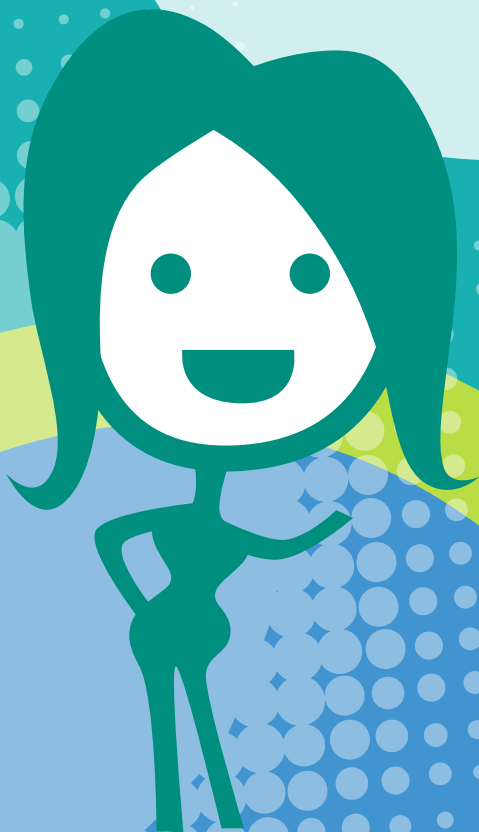


Q

Information Disclosure

Ein Angreifer kann einen Kommunikationskanal vollständig mitlesen, weil dieser unverschlüsselt ist.

Don't tell anyone, but...



K

Information Disclosure

Ein Angreifer kann Netzwerk-
informationen lesen, weil keine
Kryptografie genutzt wird.



What!*#@!
No cryptography was used?

The illustration depicts a white castle with a tall tower and a main body with battlements. A yellow speech bubble originates from the tower, containing the text 'What!*#@!' and 'No cryptography was used?'. The background is a teal gradient with a pattern of circles and dots. The entire scene is framed by a rounded rectangle.



Information Disclosure

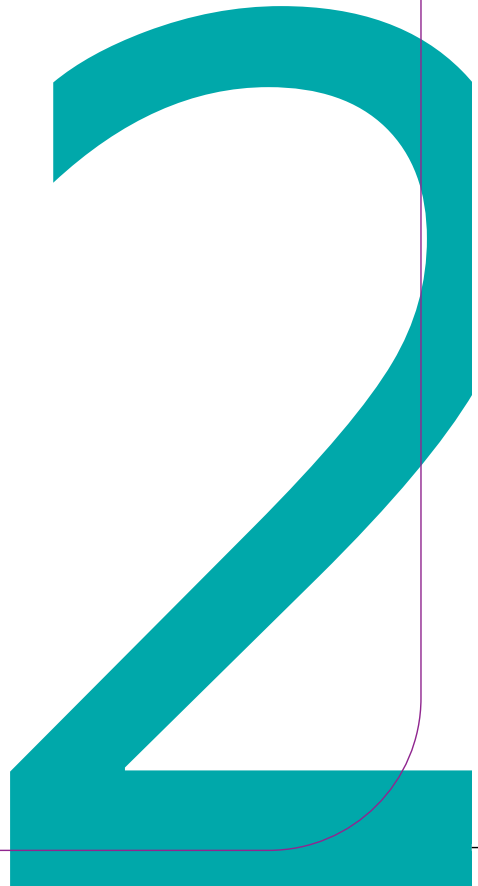
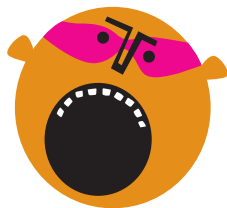
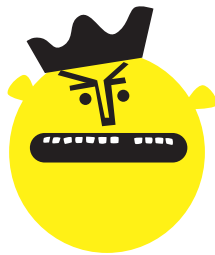
Sie haben einen neuen
Information Disclosure
Angriff erfunden.



2

Denial of Service

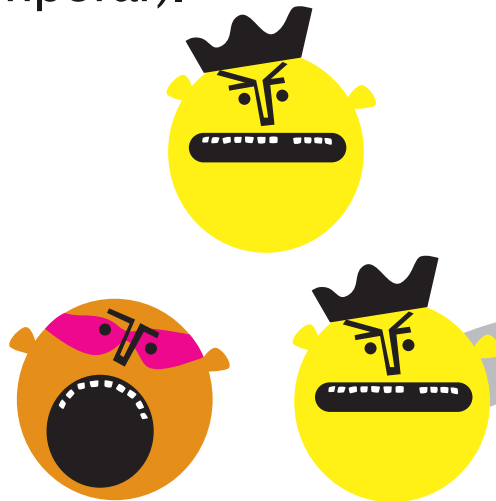
Ein Angreifer kann Ihr Authentisierungs-System unbrauchbar oder unverfügbar machen.



3

Denial of Service

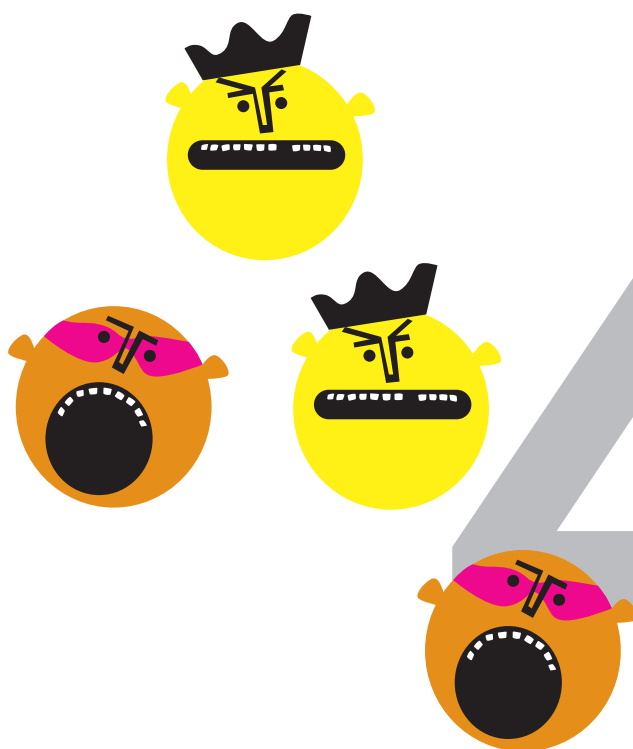
Ein Angreifer kann einen Client unverfügbar oder unbrauchbar machen, aber das Problem verschwindet, sobald der Angriff aufhört (Client, authentisiert, temporär).



4

Denial of Service

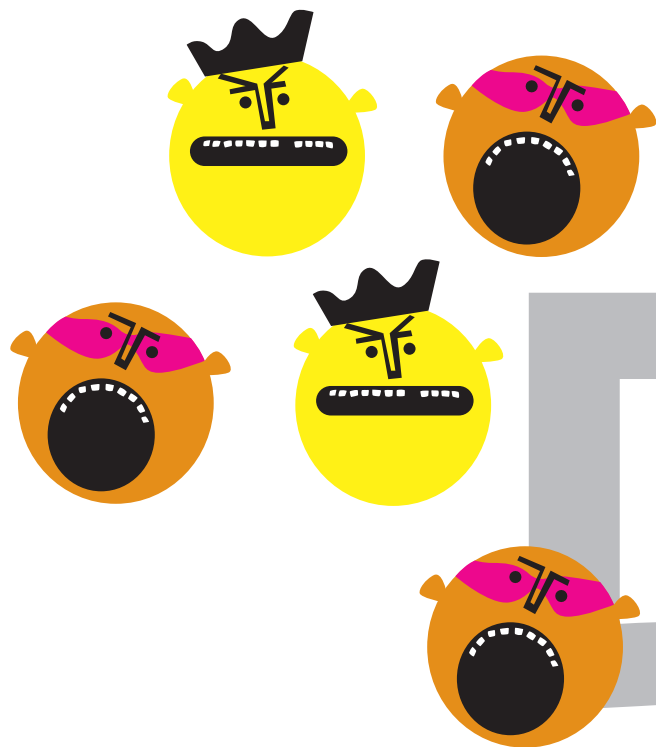
Ein Angreifer kann einen Server unverfügbar oder unbrauchbar machen, aber das Problem verschwindet, sobald der Angriff aufhört (Server, authentisiert, temporär).



5

Denial of Service

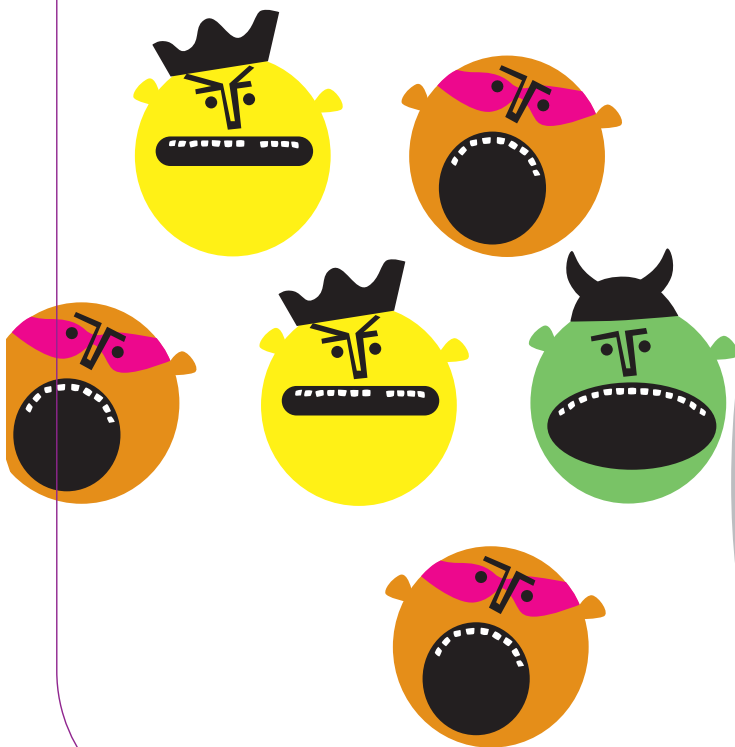
Ein Angreifer kann einen Client unverfügbar machen, ohne dass eine Authentisierung stattgefunden hat. Das Problem verschwindet nach dem Angriff (Client, anonym, temporär).



6

Denial of Service

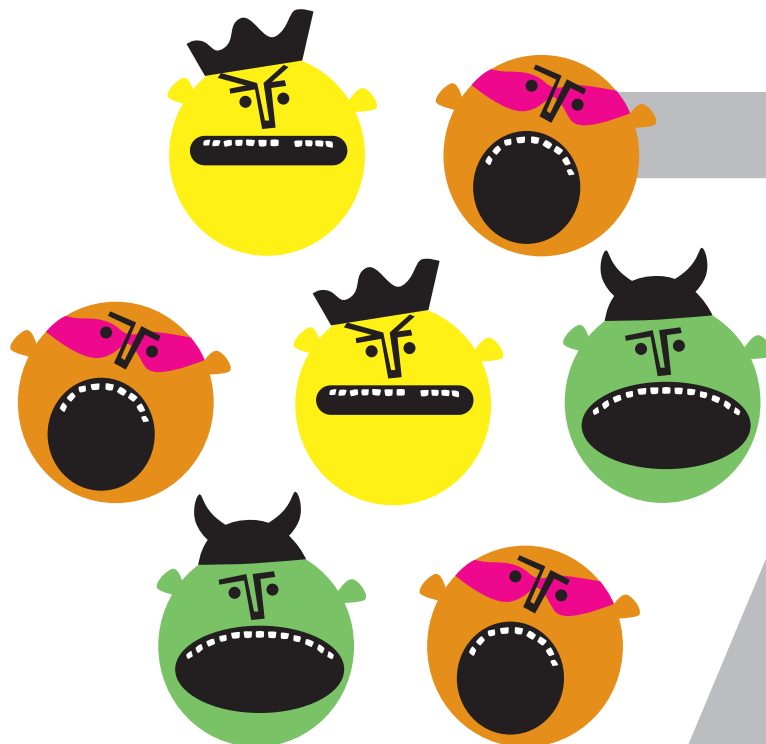
Ein Angreifer kann einen Server unverfügbar machen, ohne dass eine Authentisierung stattgefunden hat. Das Problem verschwindet nach dem Angriff (Server, anonym, temporär).



7

Denial of Service

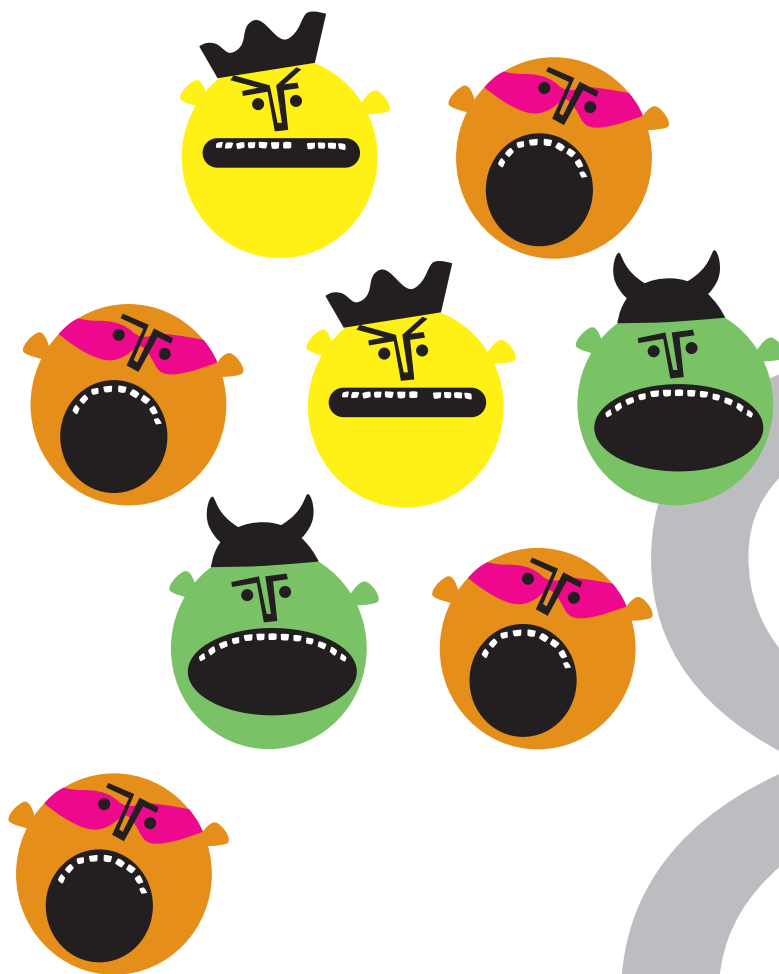
Ein Angreifer kann einen Client unverfügbar machen und das Problem besteht fort, nachdem der Angriff aufgehört hat (Client, authentisiert, persistent).



8

Denial of Service

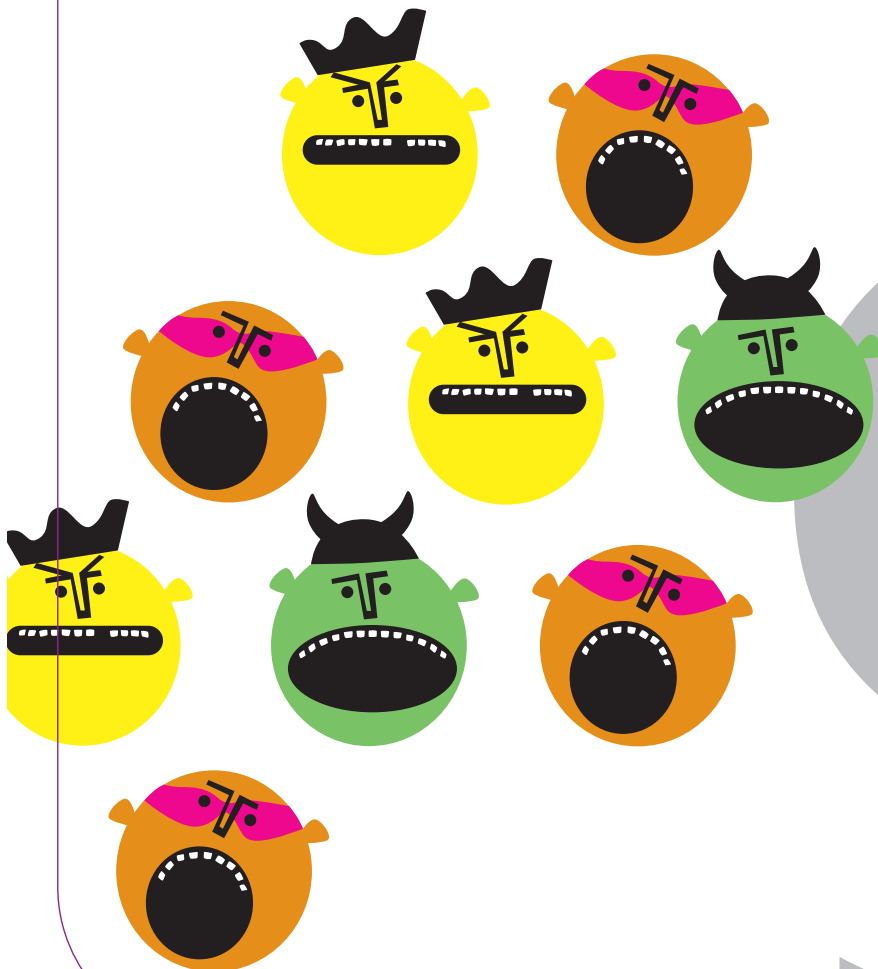
Ein Angreifer kann einen Server unverfügbar machen und das Problem besteht fort, nachdem der Angriff aufgehört hat (Server, authentisiert, persistent).



9

Denial of Service

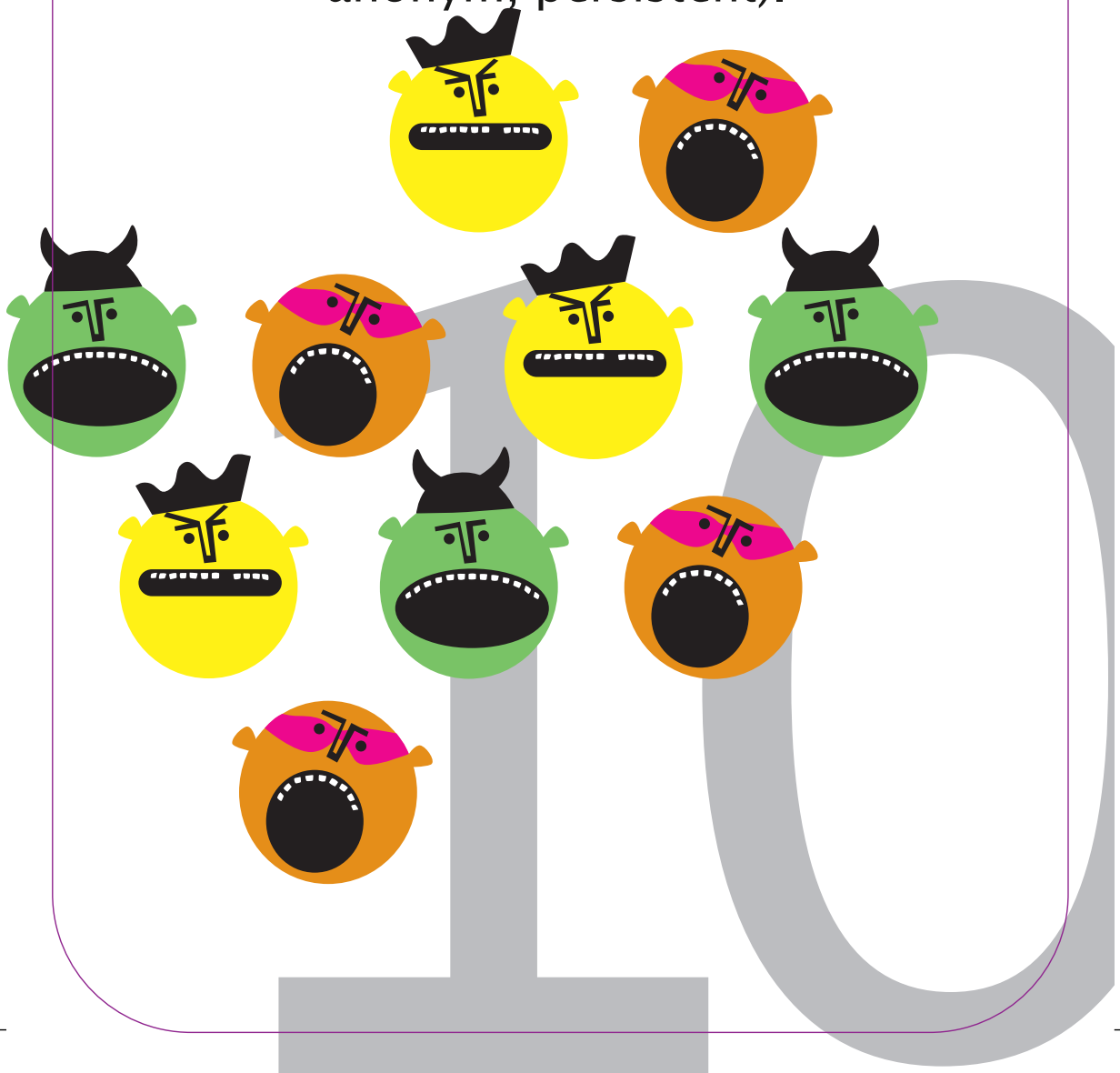
Ein Angreifer kann einen Client unverfügbar machen, ohne dass je eine Authentisierung stattgefunden hat, und das Problem besteht fort, nachdem der Angriff aufgehört hat (Client, anonym persistent).

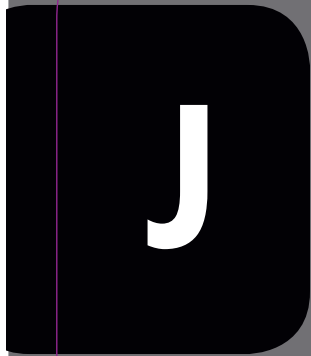


10

Denial of Service

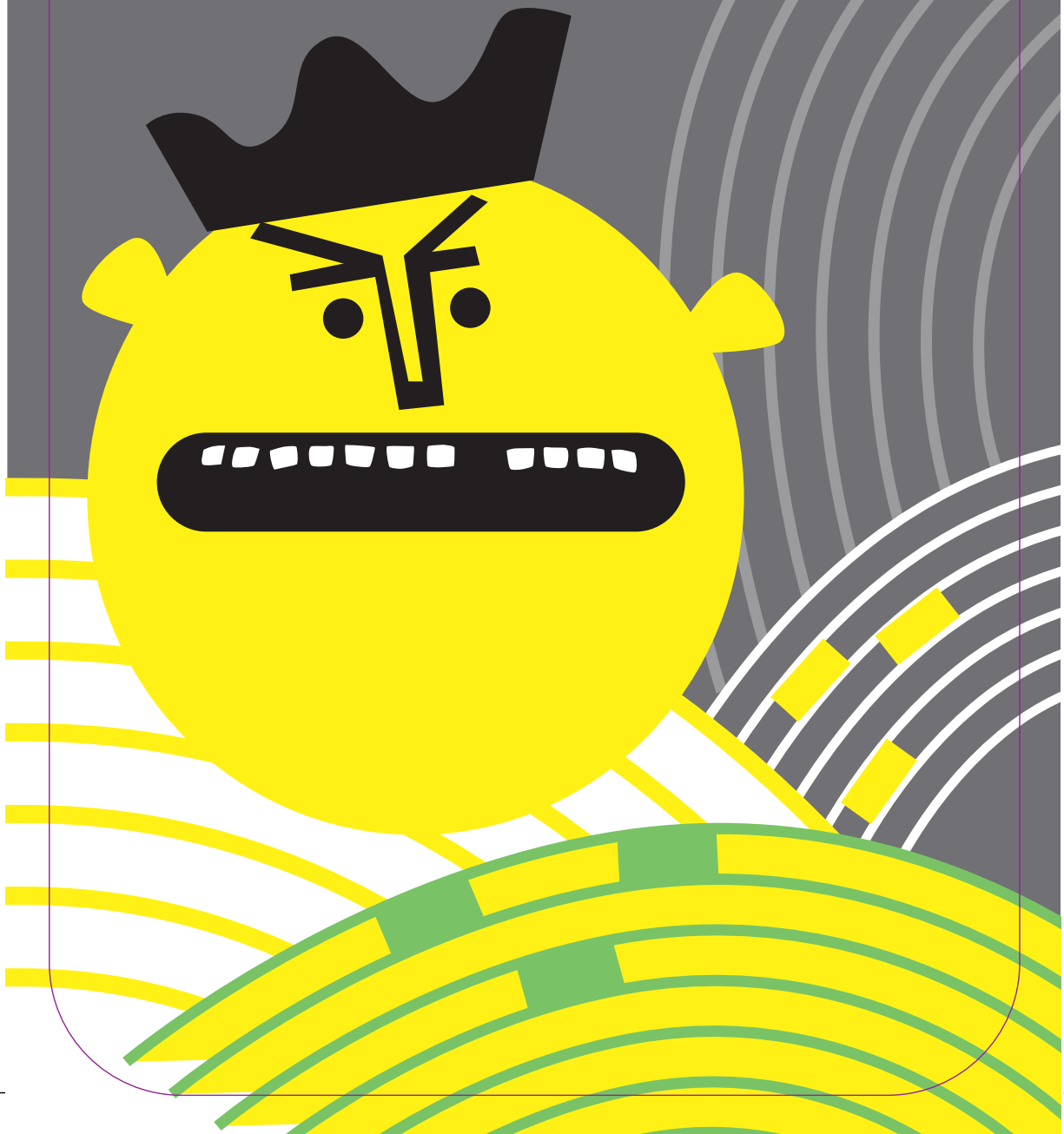
Ein Angreifer kann, ohne zu authentisieren, einen Server un-
verfügbar machen. Das Problem
besteht fort, nachdem der An-
griff aufgehört hat (Server,
anonym, persistent).





Denial of Service

Ein Angreifer kann das Logging-Subsystem außer Betrieb setzen.



Q

Denial of Service

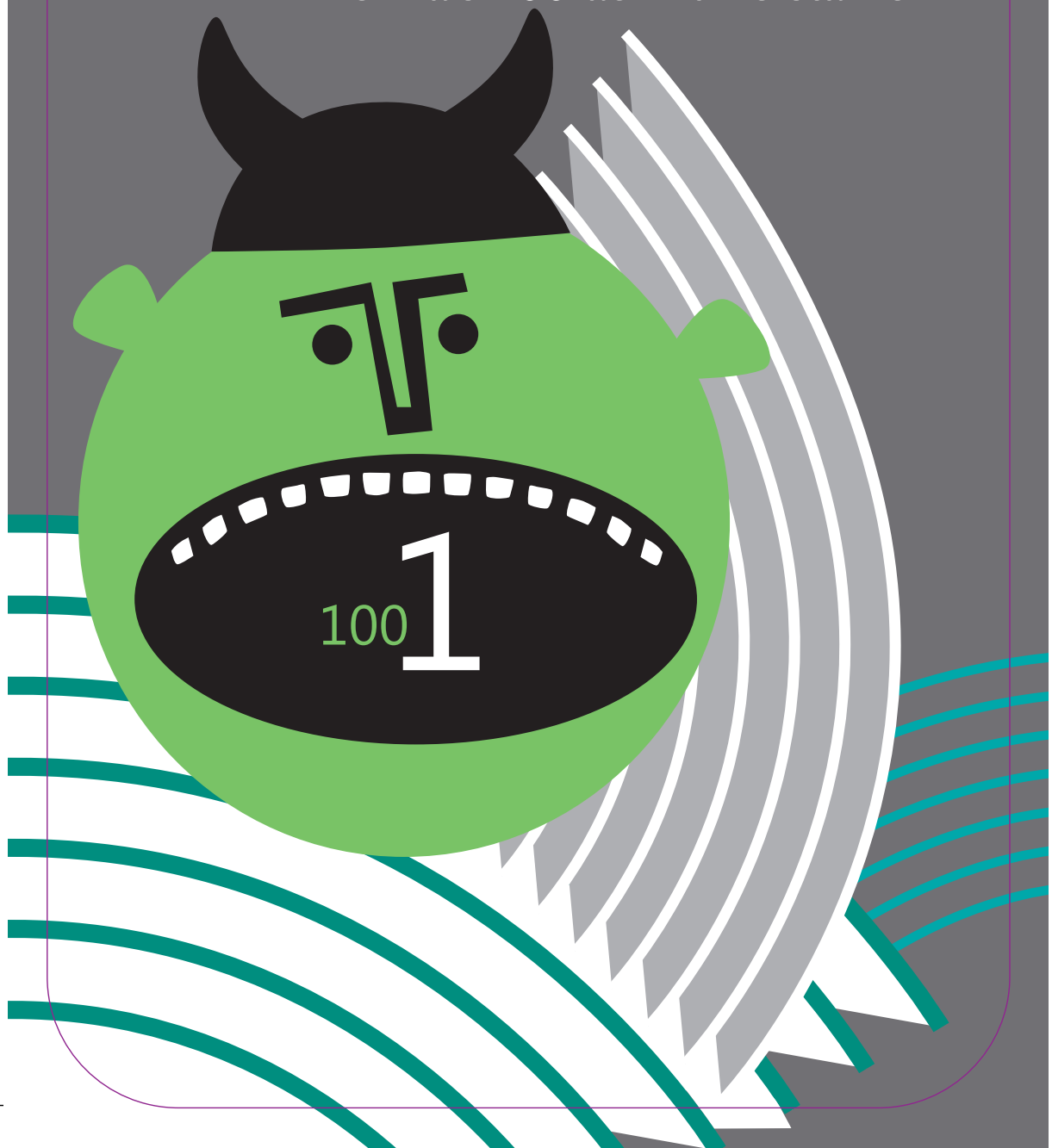
Ein Angreifer kann eine verwundbare Systemkomponente missbrauchen, um eine (volumenbasierte) DoS Attacke um den Faktor 10 zu verstärken.

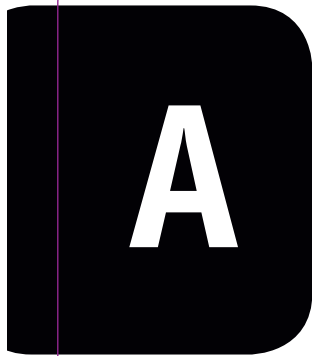


K

Denial of Service

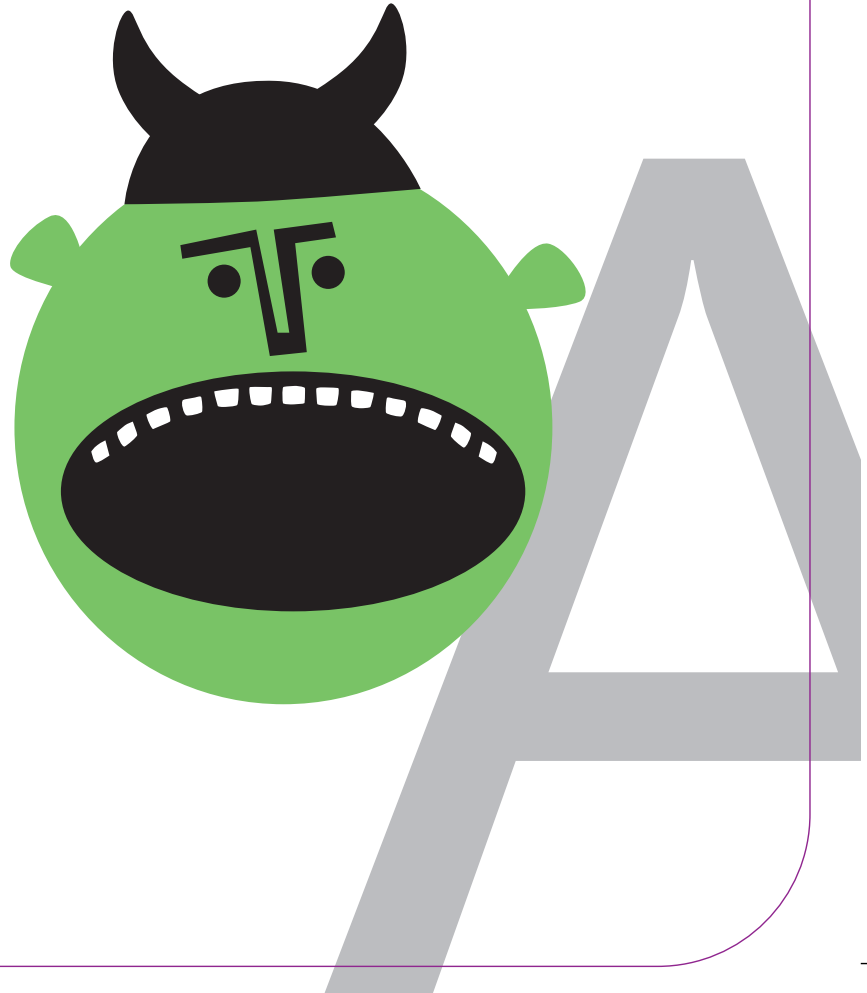
Ein Angreifer kann eine verwundbare Systemkomponente missbrauchen, um eine DoS Attacke mehr als 100fach zu verstärken.





Denial of Service

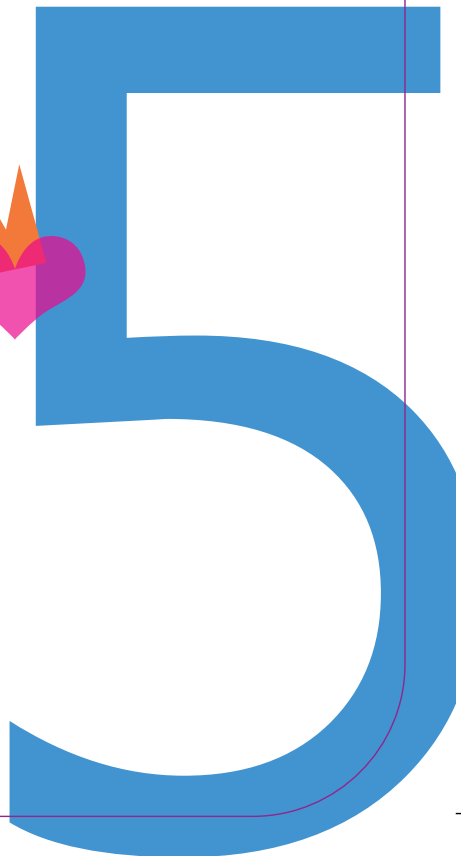
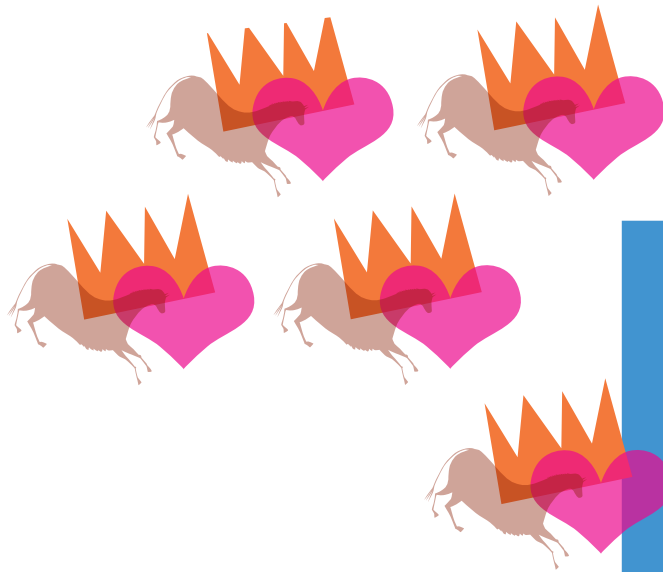
Sie haben eine neue Denial of Service Attacke erfunden.



5

Elevation of Privilege

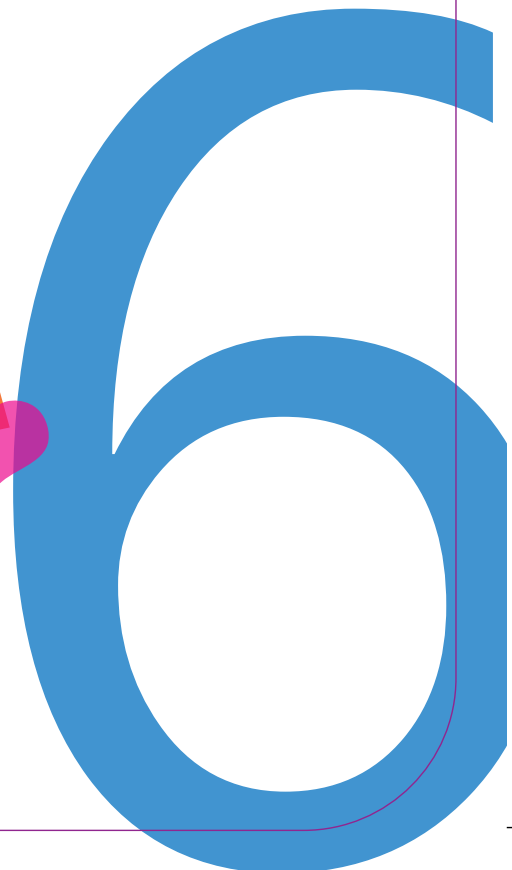
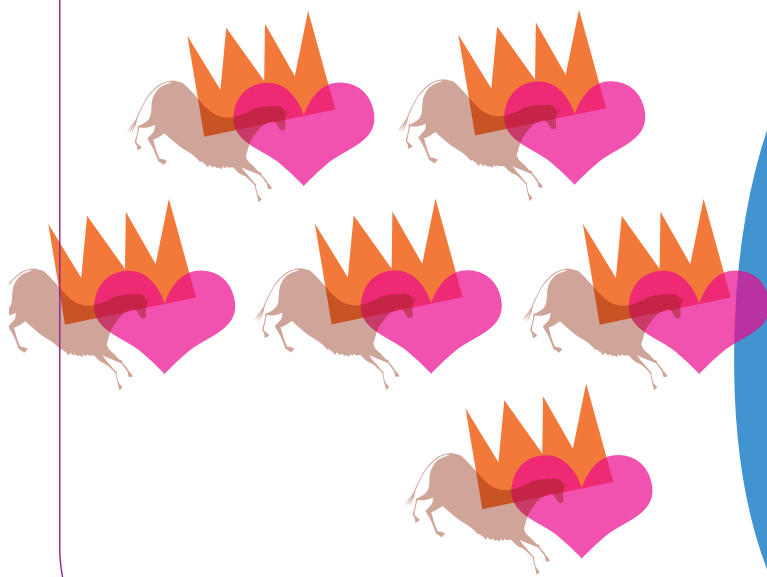
Ein Angreifer kann Einfluss darauf nehmen, welche Art Validierung Daten durchlaufen, die jeweils unterschiedliche Ergebnisse liefern.



6

Elevation of Privilege

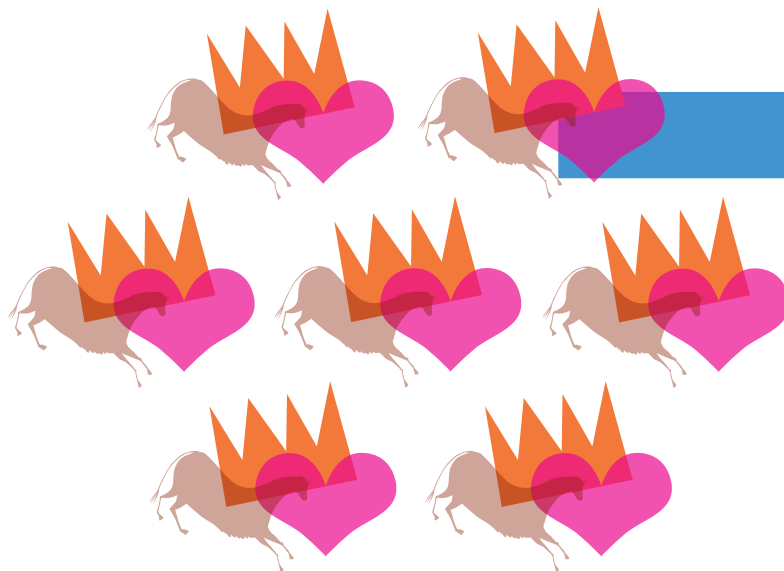
Ein Angreifer kann Berechtigungen, die Ihr Programm verlangt, aber nicht wirklich benötigt, für sich ausnutzen.



7

Elevation of Privilege

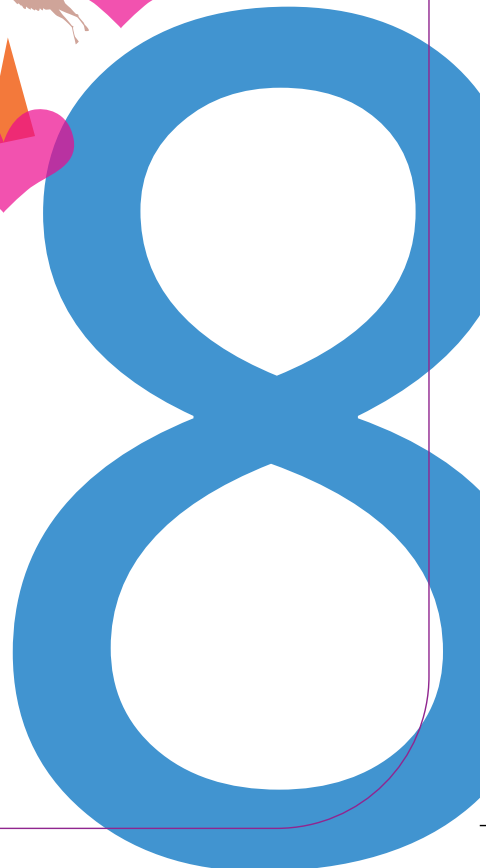
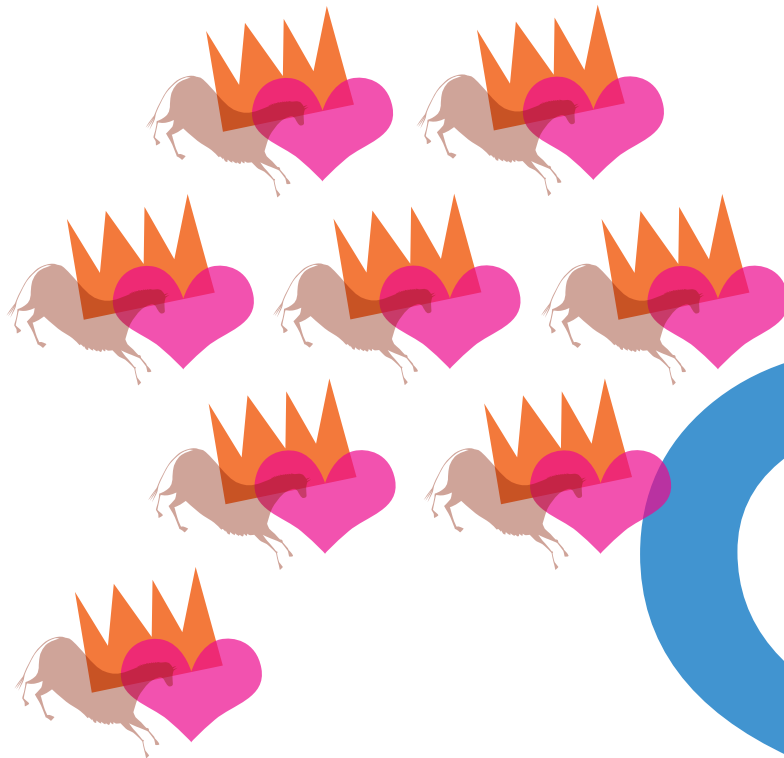
Ein Angreifer kann einen Pointer über eine Trust-Boundary hinweg angeben, anstatt Daten eingeben zu müssen, die eine Validierung durchlaufen.



8

Elevation of Privilege

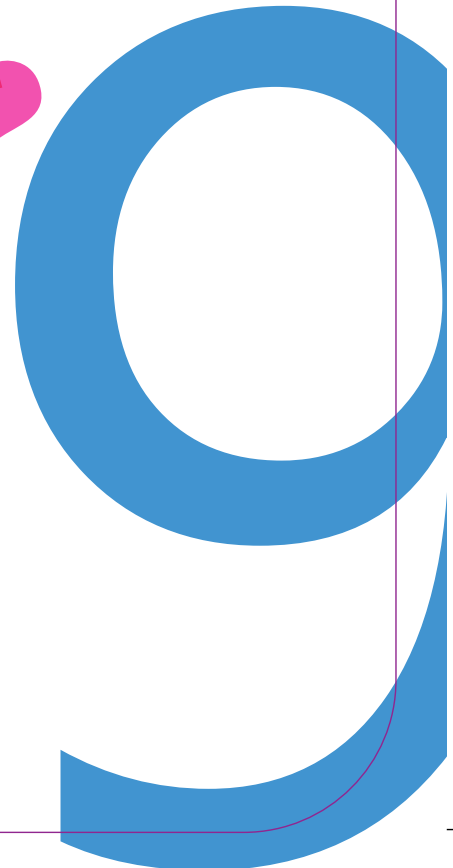
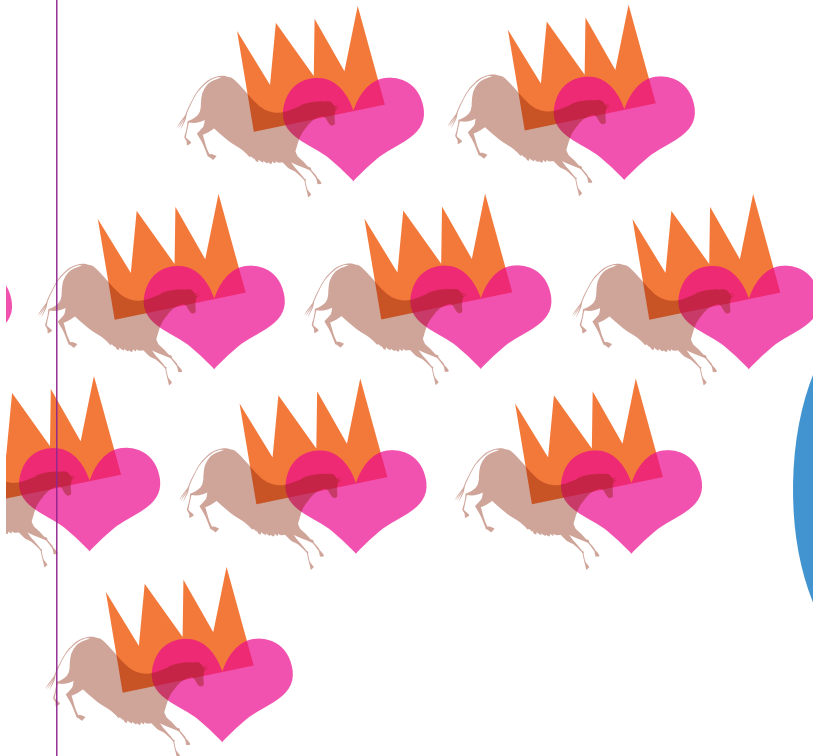
Eingegebene Daten befinden sich während der Überprüfung noch unter Kontrolle des Angreifers und werden später jenseits der Trust Boundary genutzt.s



9

Elevation of Privilege

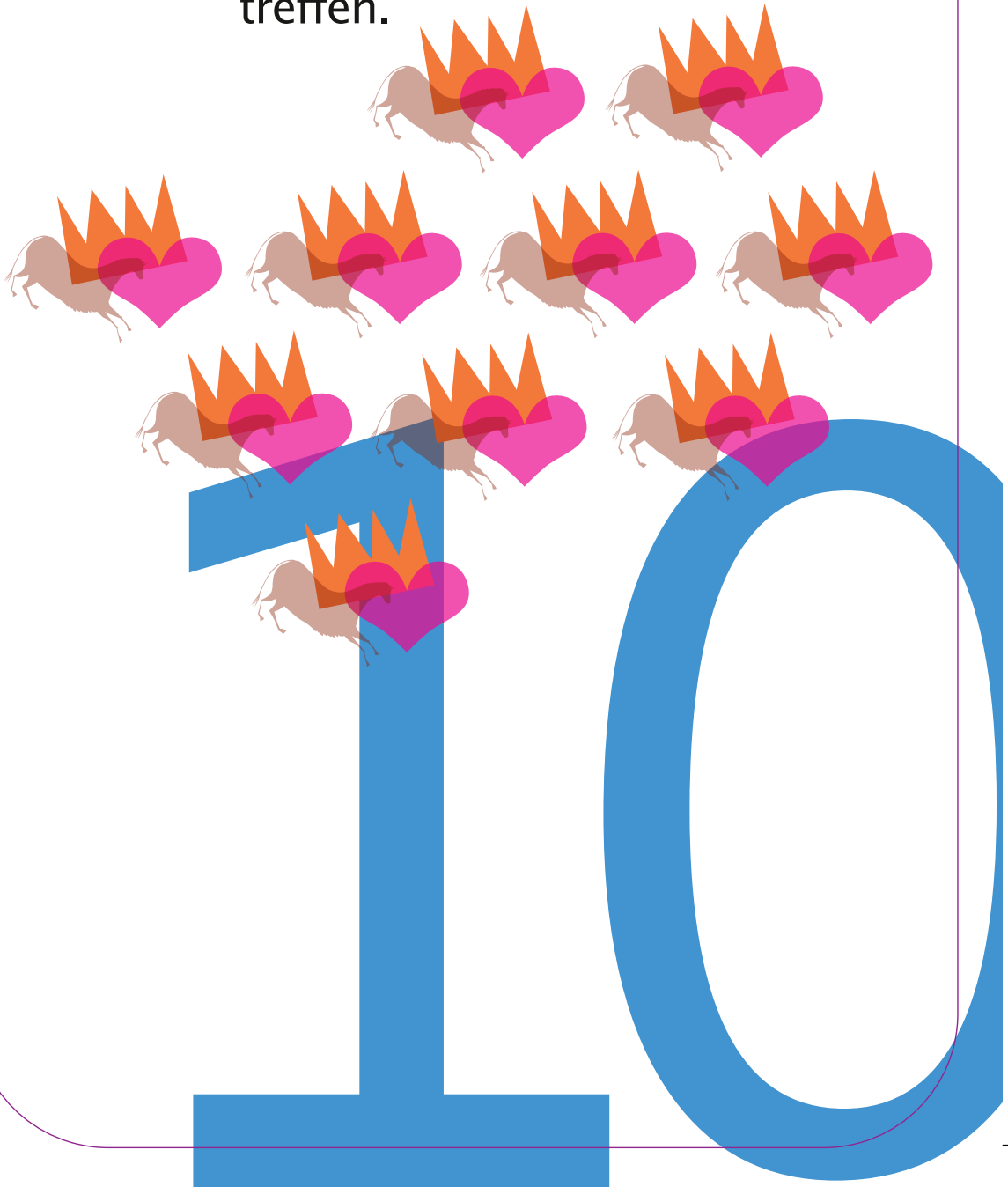
Caller (aufrufende Funktionen) haben keine Möglichkeit zu überprüfen, welche Validierung Ihr Programm auf die übergebenen Daten anwendet, bevor sie diese weitergeben.



10

Elevation of Privilege

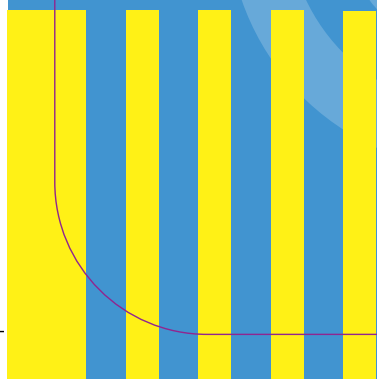
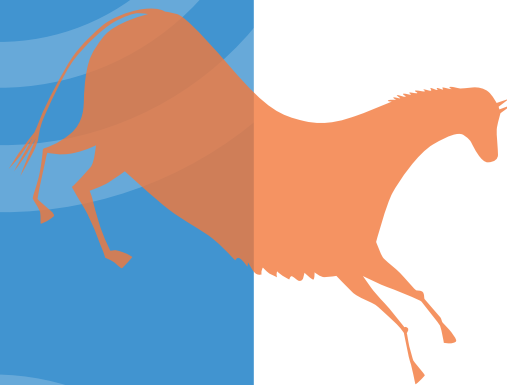
Es ist für einen Aufrufer/Caller
nicht klar ersichtlich, welche
Sicherheitsannahmen Sie
treffen.





Elevation of Privilege

Ein Angreifer kann Eingaben zum Nutzer zurückspiegeln, z.B. per Cross-Site-Scripting (XSS).





Q

Elevation


Sie inkludieren benutzergenerierten Inhalt (User Generated Content, UGC) in Ihrer Webseite, der auch Inhalte beliebiger URLs etc. enthalten kann.



K

Elevation of Privilege

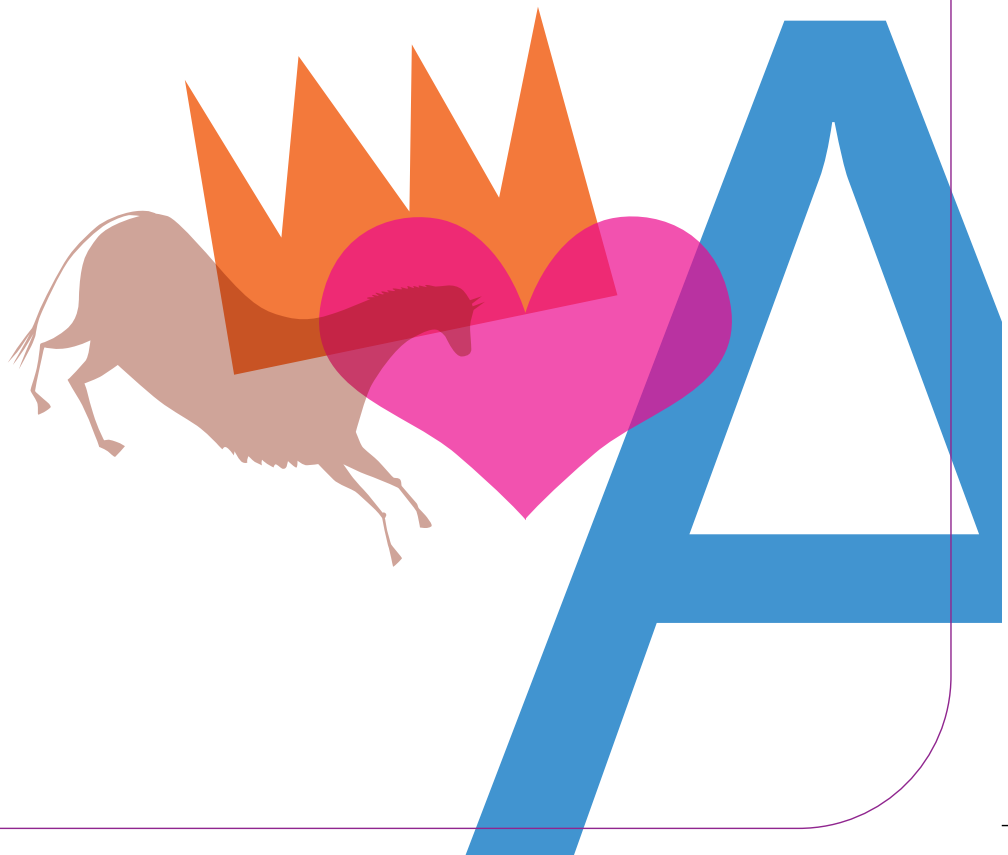
Ein Angreifer kann ein Kommando einschleusen, welches vom System mit einer höheren Berechtigung ausgeführt wird.



A

Elevation of Privilege

Sie haben einen neuen
Elevation of Privilege Angriff
erfunden.





Spoofing

2. An attacker could squat on the random port or socket that the server normally uses.
3. An attacker could try one credential after another and there's nothing to slow them down (online or offline).
4. An attacker can anonymously connect because we expect authentication to be done at a higher level.
5. An attacker can confuse a client because there are too many ways to identify a server.
6. An attacker can spoof a server because identifiers aren't stored on the client and checked for consistency on re-connection (that is, there's no key persistence).
7. An attacker can connect to a server or peer over a link that isn't authenticated (and encrypted).
8. An attacker could steal credentials stored on the server and reuse them (for example, a key is stored in a world readable file).
9. An attacker who gets a password can reuse it (use stronger authenticators).

continued on back

Spoofing



Tampering

3. An attacker can take advantage of your custom key exchange or integrity control which you built instead of using standard crypto.
4. Your code makes access control decisions all over the place, rather than with a security kernel.
5. An attacker can replay data without detection because your code doesn't provide timestamps or sequence numbers.
6. An attacker can write to a data store your code relies on.
7. An attacker can bypass permissions because you don't make names canonical before checking access permissions.
8. An attacker can manipulate data because there's no integrity protection for data on the network.
9. An attacker can provide or control state information.
10. An attacker can alter information in a data store because it has weak ACLs or includes a group which is equivalent to everyone ("all Live ID holders").

continued on back

Tampering



R

Repudiation

2. An attacker can pass data through the log to attack a log reader, and there's no documentation of what sorts of validation are done.
3. A low privilege attacker can read interesting security information in the logs.
4. An attacker can alter files or messages because the digital signature system you're implementing is weak, or uses MACs where it should use a signature.
5. An attacker can alter log messages on a network because they lack strong integrity controls.
6. An attacker can create a log entry without a timestamp (or no log entry is timestamped).
7. An attacker can make the logs wrap around and lose data.
8. An attacker can make a log lose or confuse security information.
9. An attacker can use a shared key to authenticate as different principals, confusing the information in the logs.

continued on back

Repudiation



Information Disclosure

2. An attacker can brute-force file encryption because there's no defense in place (example defense: password stretching).
3. An attacker can see error messages with security-sensitive content.
4. An attacker can read content because messages (for example, an email or HTTP cookie) aren't encrypted even if the channel is encrypted.
5. An attacker may be able to read a document or data because it's encrypted with a non-standard algorithm.
6. An attacker can read data because it's hidden or occluded (for undo or change tracking) and the user might forget that it's there.
7. An attacker can act as a "man in the middle" because you don't authenticate endpoints of a network connection.
8. An attacker can access information through a search indexer, logger, or other such mechanism.

continued on back

Information Disclosure

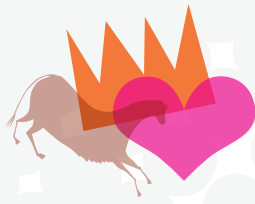


Denial of Service

2. An attacker can make your authentication system unusable or unavailable.
3. An attacker can make a client unavailable or unusable but the problem goes away when the attacker stops **(client, authenticated, temporary)**.
4. An attacker can make a server unavailable or unusable but the problem goes away when the attacker stops **(server, authenticated, temporary)**.
5. An attacker can make a client unavailable or unusable without ever authenticating, but the problem goes away when the attacker stops **(client, anonymous, temporary)**.
6. An attacker can make a server unavailable or unusable without ever authenticating, but the problem goes away when the attacker stops **(server, anonymous, temporary)**.
7. An attacker can make a client unavailable or unusable and the problem persists after the attacker goes away **(client, authenticated, persistent)**.

continued on back

Denial of Service



Elevation of Privilege (EoP)

- 5.** An attacker can force data through different validation paths which give different results.
- 6.** An attacker could take advantage of .NET permissions you ask for, but don't use.
- 7.** An attacker can provide a pointer across a trust boundary, rather than data which can be validated.
- 8.** An attacker can enter data that is checked while still under the attacker's control and used later on the other side of a trust boundary.
- 9.** There's no reasonable way for callers to figure out what validation of tainted data you perform before passing it to them.
- 10.** There's no reasonable way for a caller to figure out what security assumptions you make.
- J.** An attacker can reflect input back to a user, like cross-site scripting.
- Q.** You include user-generated content within your page, possibly including the content of random URLs.
- K.** An attacker can inject a command that the system will run at a higher privilege level.
- A.** You've invented a new Elevation of Privilege attack.

Elevation of Privilege



About

Threat Modeling

The Elevation of Privilege game is designed to be the easiest way to start looking at your design from a security perspective. It's one way to threat model, intended to be picked up and used by any development group. Because the game uses STRIDE threats, it gives you a framework for thinking, and specific actionable examples of those threats.

STRIDE stands for:

Spoofing: Impersonating something or someone else.

Tampering: Modifying data or code.

Repudiation: Claiming not to have performed an action.

Information Disclosure: Exposing information to someone not authorized to see it.

Denial of Service: Denying or degrading service to users.

Elevation of Privilege: Gain capabilities without proper authorization.

At www.microsoft.com/security/sdl/eop we have videos, score sheets and tips and tricks for playing.

About