



Anleitung 1

Elevation of Privilege Anleitung

1. Zeichnen Sie ein Diagramm des Systems, für das Sie ein Bedrohungsmoell 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 Punktzahlen 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.

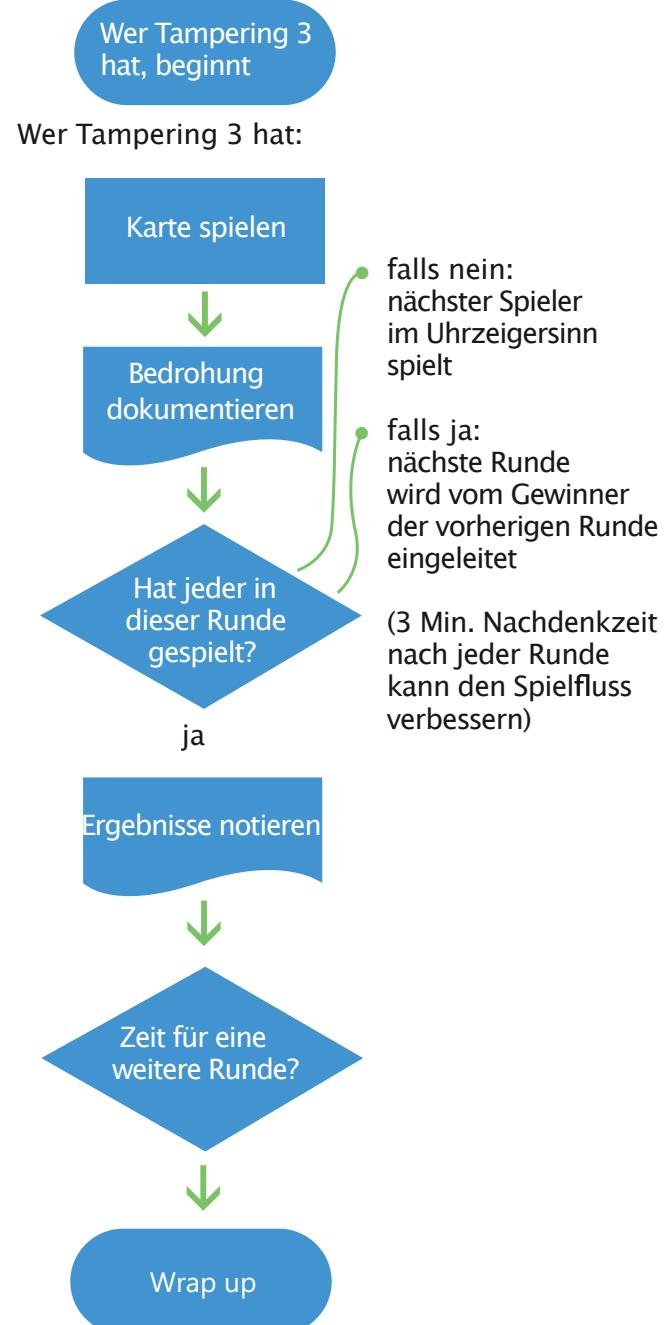
Thanks to Laurie Williams for inspiration.

Anleitung

Context



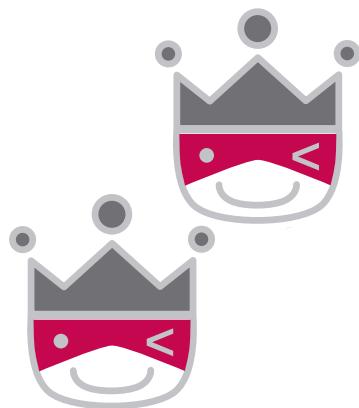
Play



2

Spoofing

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



3

Spoofing

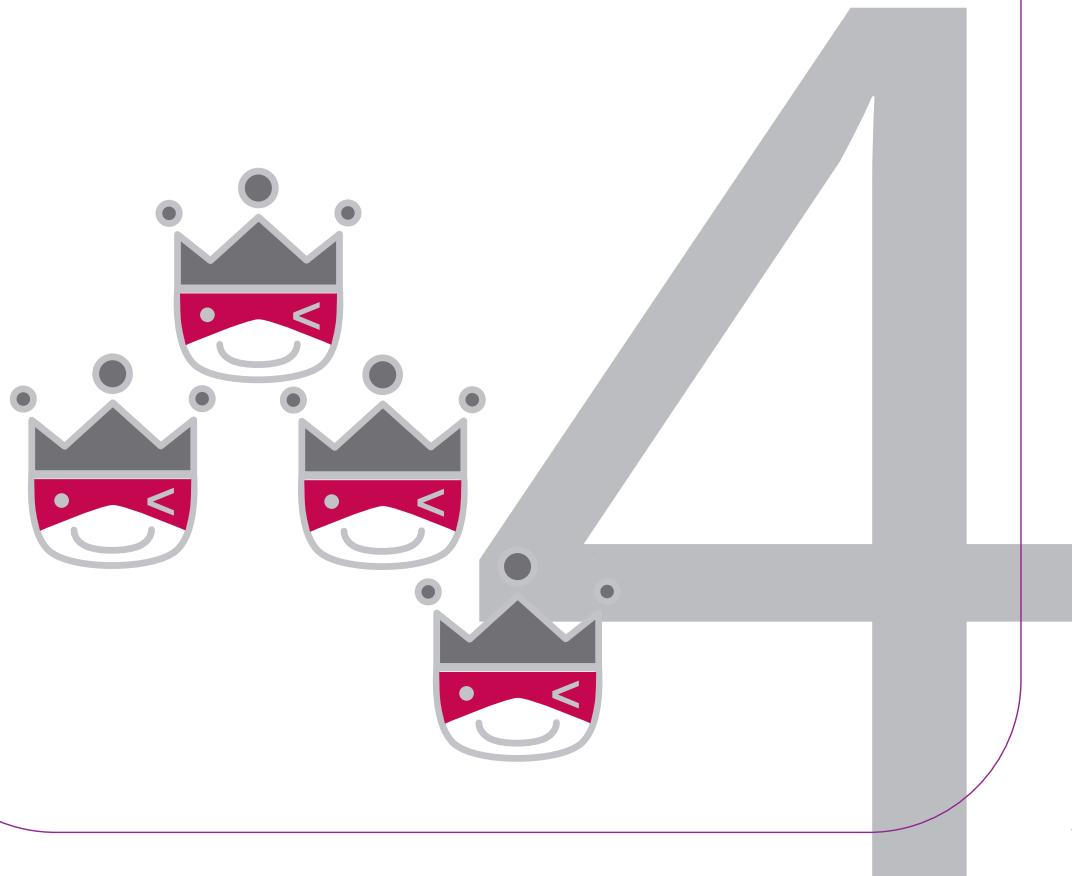
Ein Angreifer kann alle mögl. Credentials der Reihe nach durchprobieren und es gibt keinen Mechanismus, der ihn ausbremst (online oder offline).



4

Spoofing

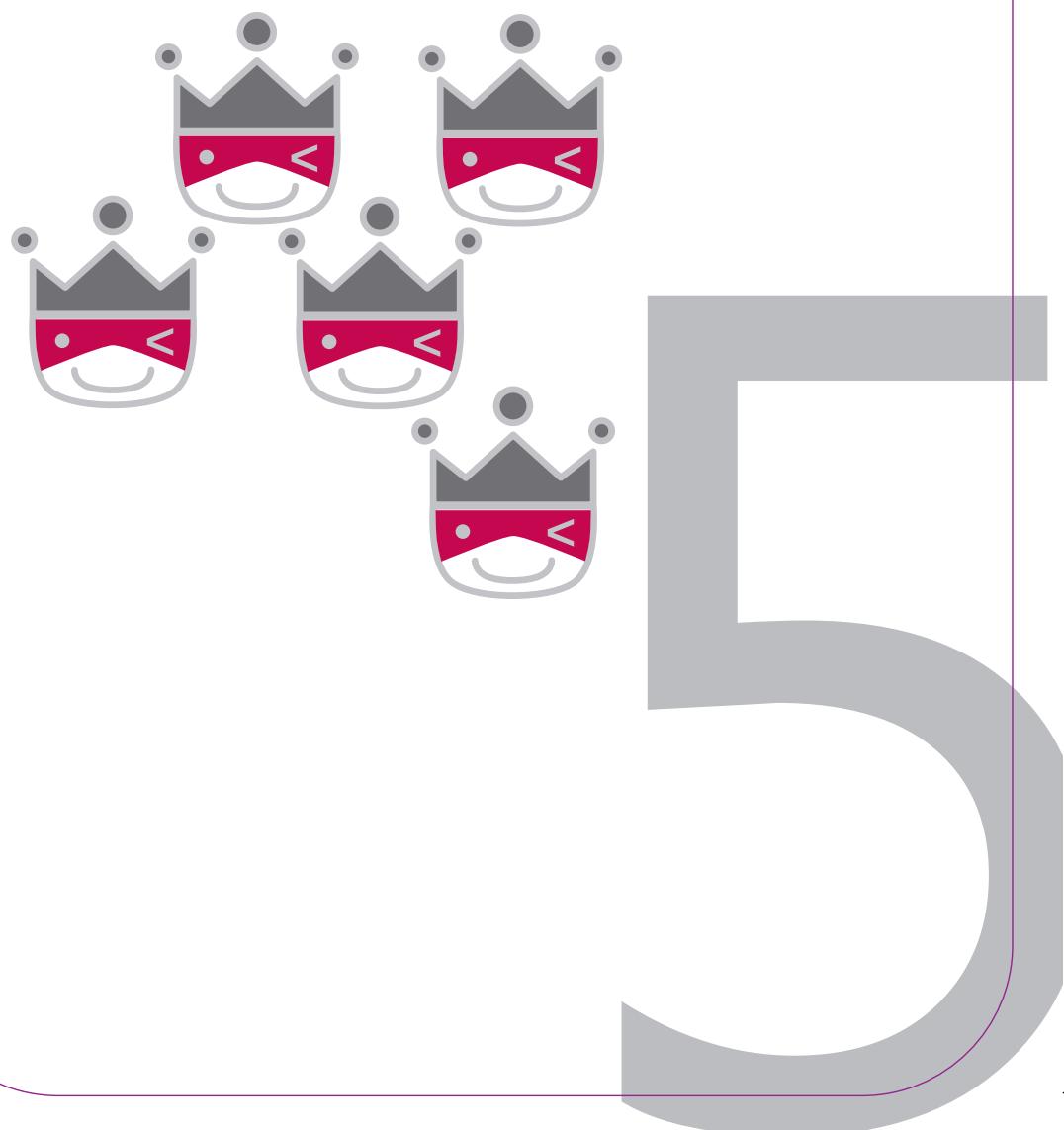
Ein Angreifer kann sich anonym verbinden, weil Sie davon ausgehen, dass Authentisierung auf einer höheren Schicht stattfindet.



5

Spoofing

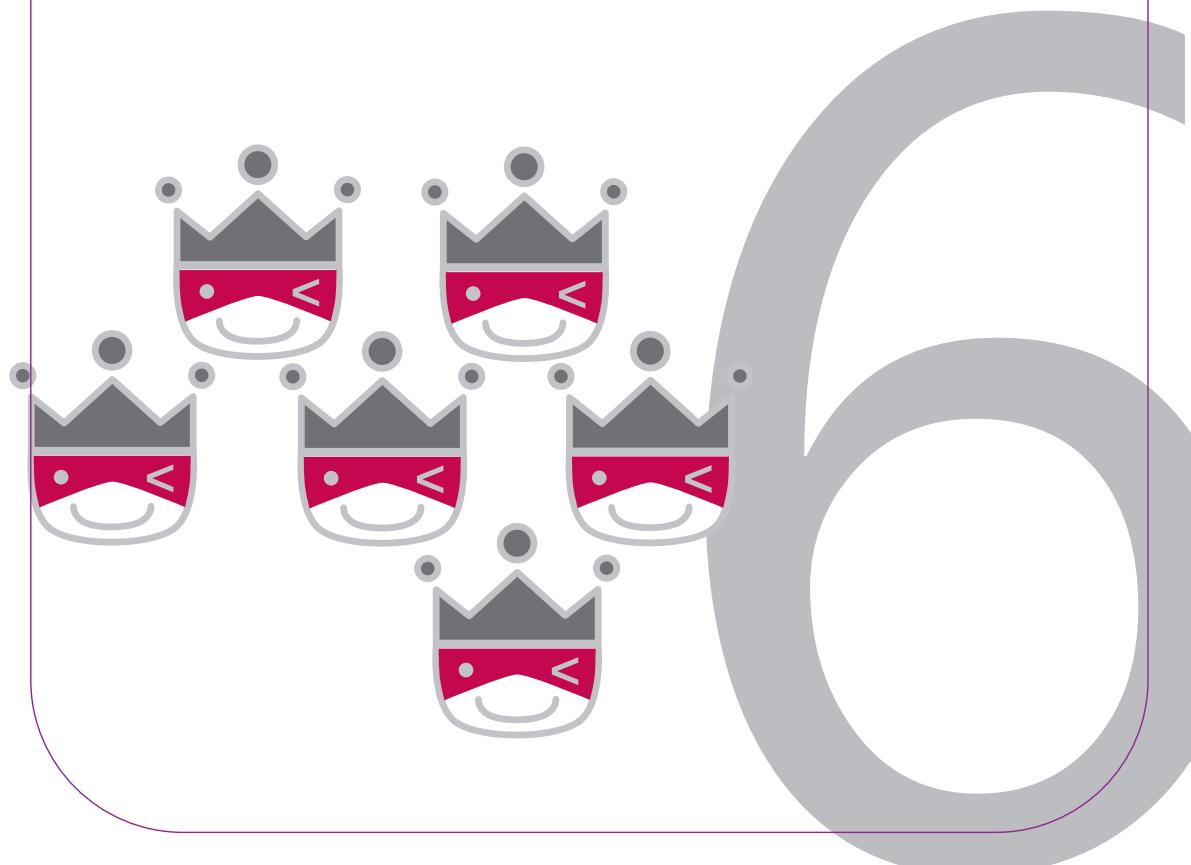
Ein Angreifer kann einen Client verwirren, weil es zu viele Wege gibt, einen Server zu identifizieren.



6

Spoofing

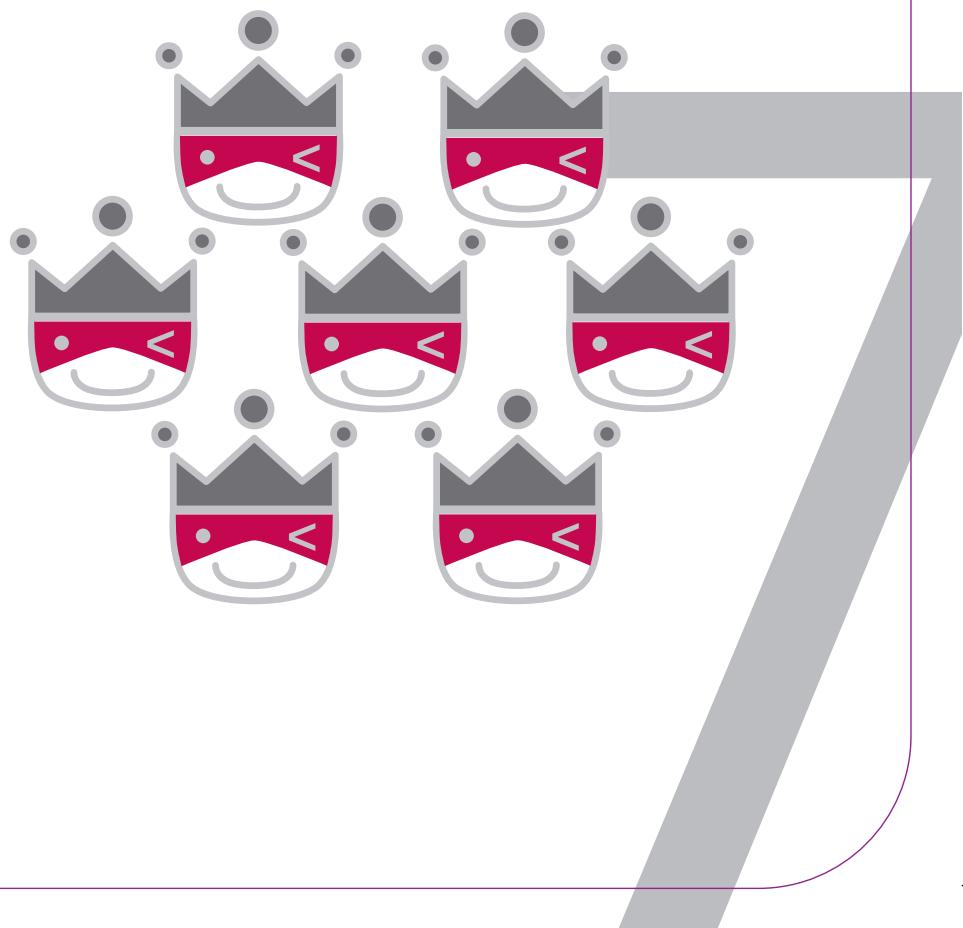
Ein Angreifer kann einen Server spoofen, weil auf dem Client keinerlei Identifizierungsmerkmale gespeichert sind, die bei erneuter Verbindung überprüft würden (es gibt keine Key persistence).



7

Spoofing

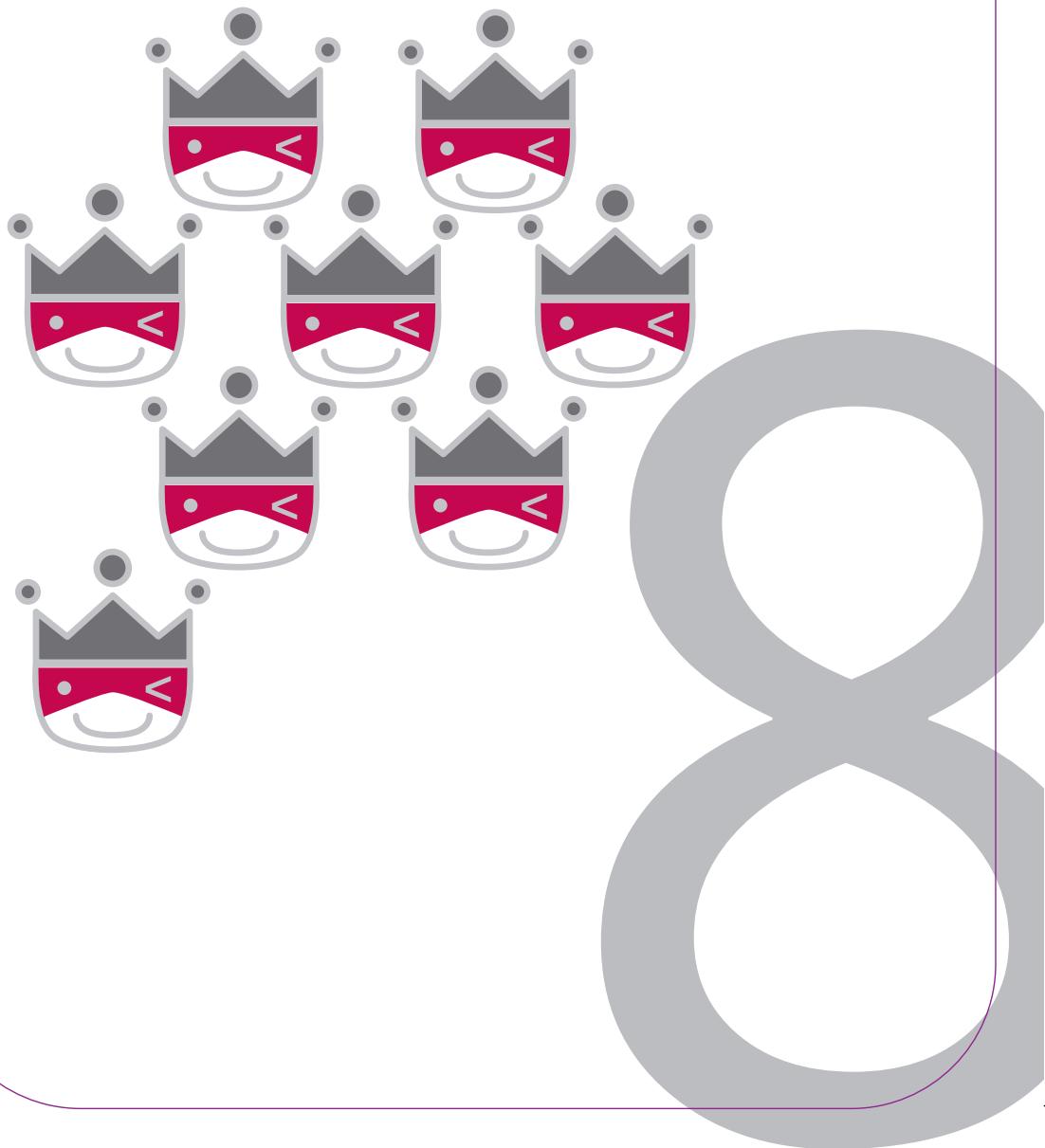
Ein Angreifer kann sich zu einem Server oder Peer über einen nicht authentisierten unverschlüsselten Kanal verbinden.



8

Spoofing

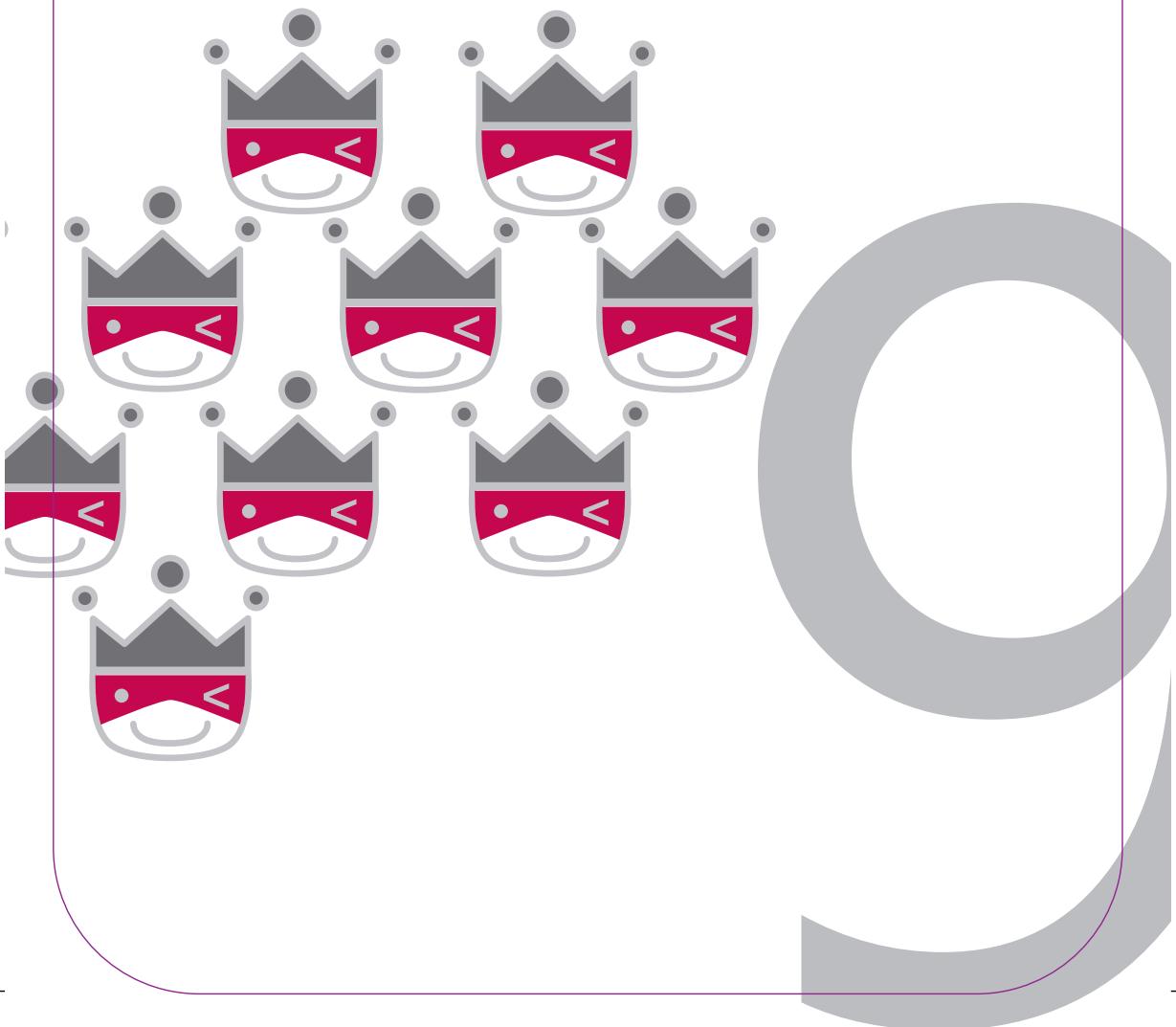
Ein Angreifer kann auf einem Server gespeicherte Credentials stehlen und wieder verwenden (z.B. Schlüssel in einer für andere lesbaren Datei).



9

Spoofing

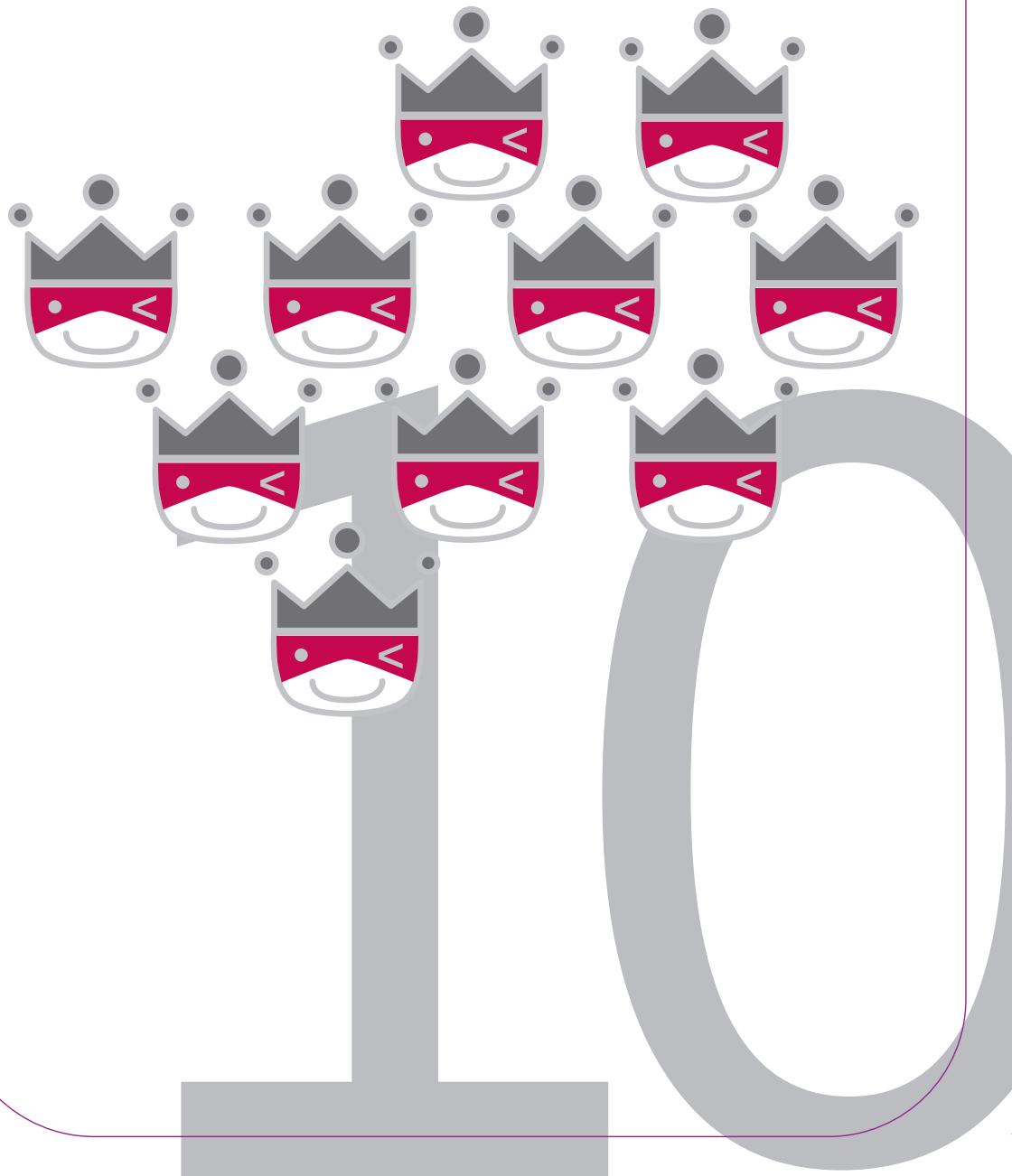
Ein Angreifer, der Zugang zu einem Passwort bekommt, kann es wieder verwenden (nutzen Sie stärkere Authentisierungsmethoden).



10

Spoofing

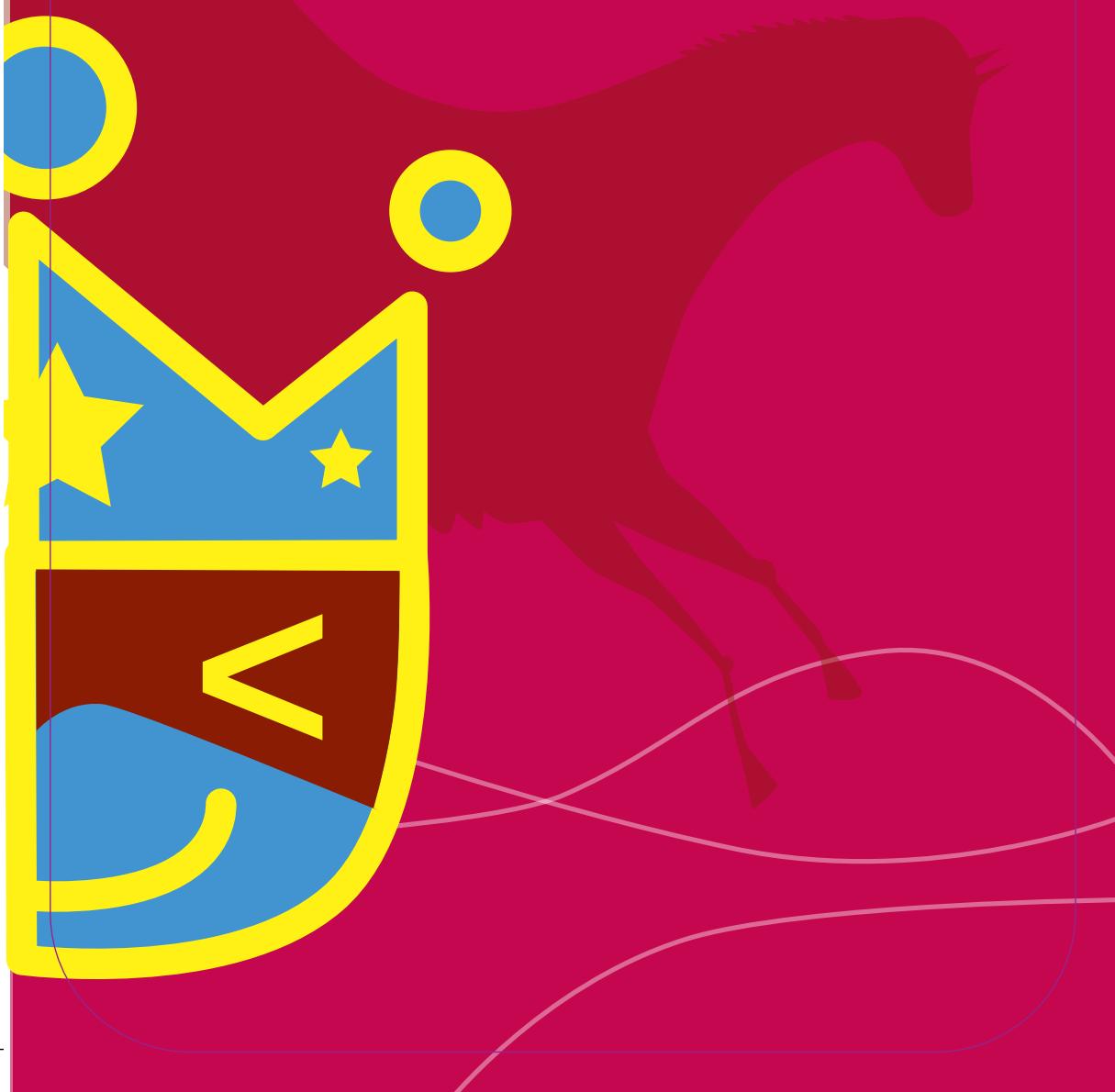
Ein Angreifer kann wählen, dass eine schwächere oder gar keine Authentisierung genutzt wird.



J

Spoofing

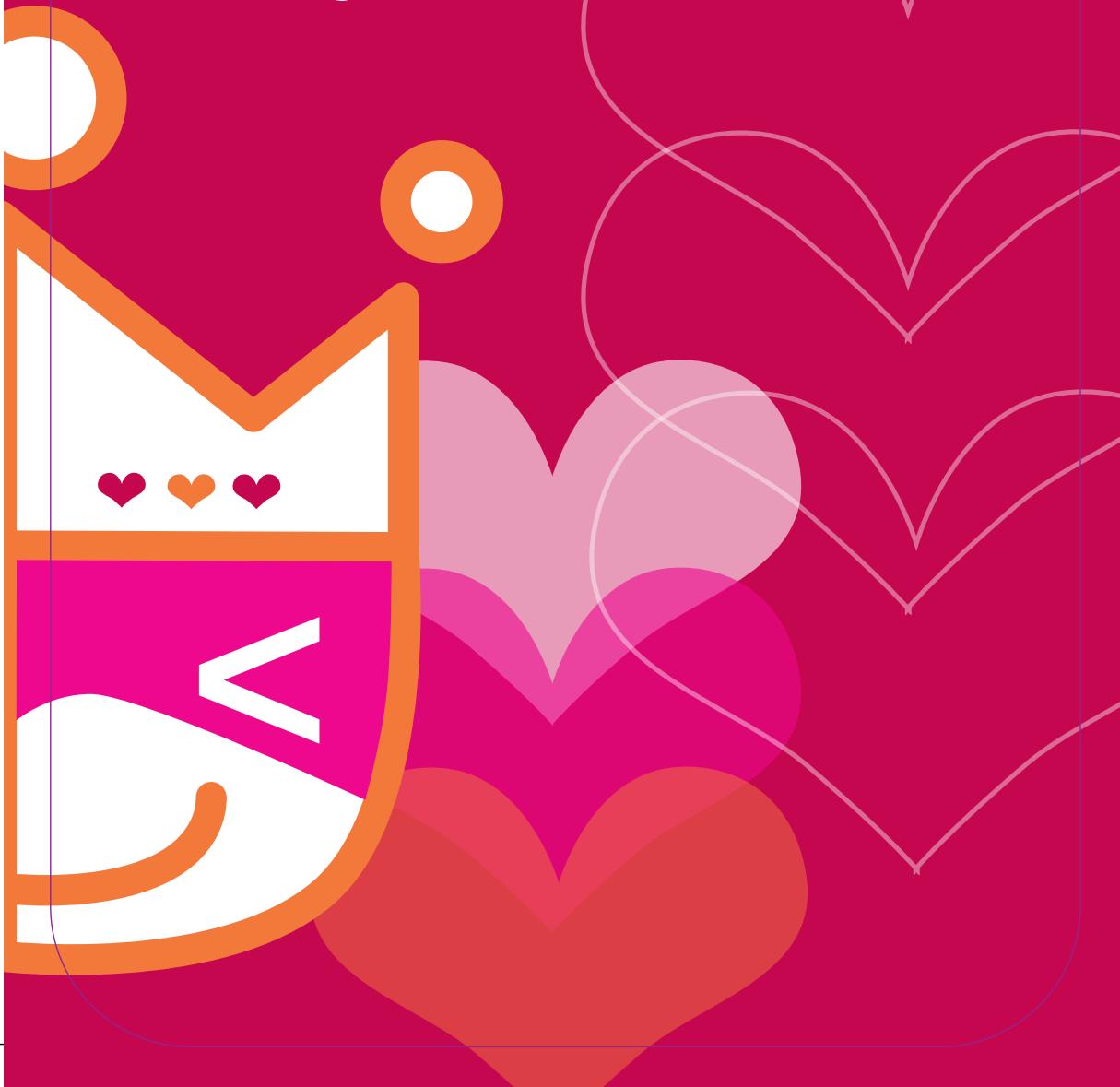
Ein Angreifer kann die auf einem Client gespeicherten Credentials stehlen und wieder verwenden.



Q

Spoofing

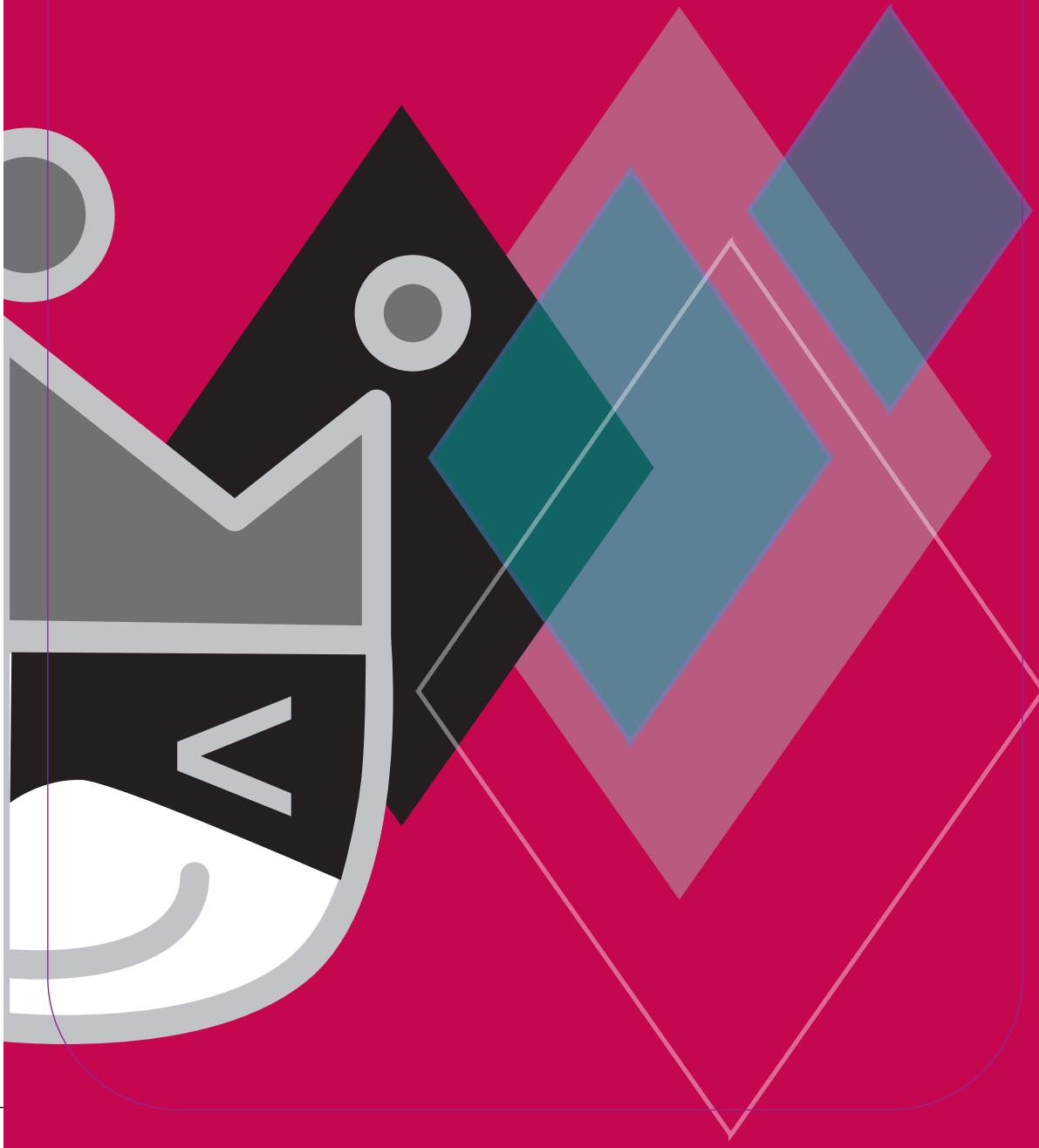
Ein Angreifer kann den Mechanismus angreifen, mit dem Passwörter zurückgesetzt oder aktualisiert werden (Account Recovery erfordert nicht die Eingabe des alten Passworts).



K

Spoofing

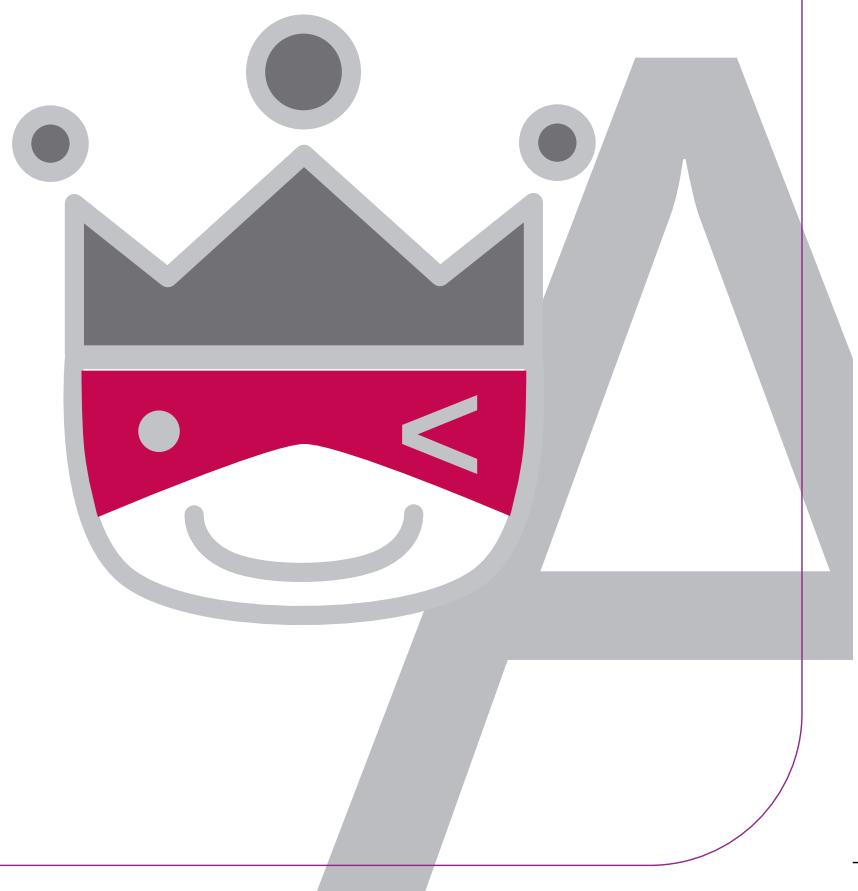
Ihr System wird mit einem Default Adminpasswort ausgeliefert und erzwingt nicht die Änderung dieses Passworts.



A

Spoofing

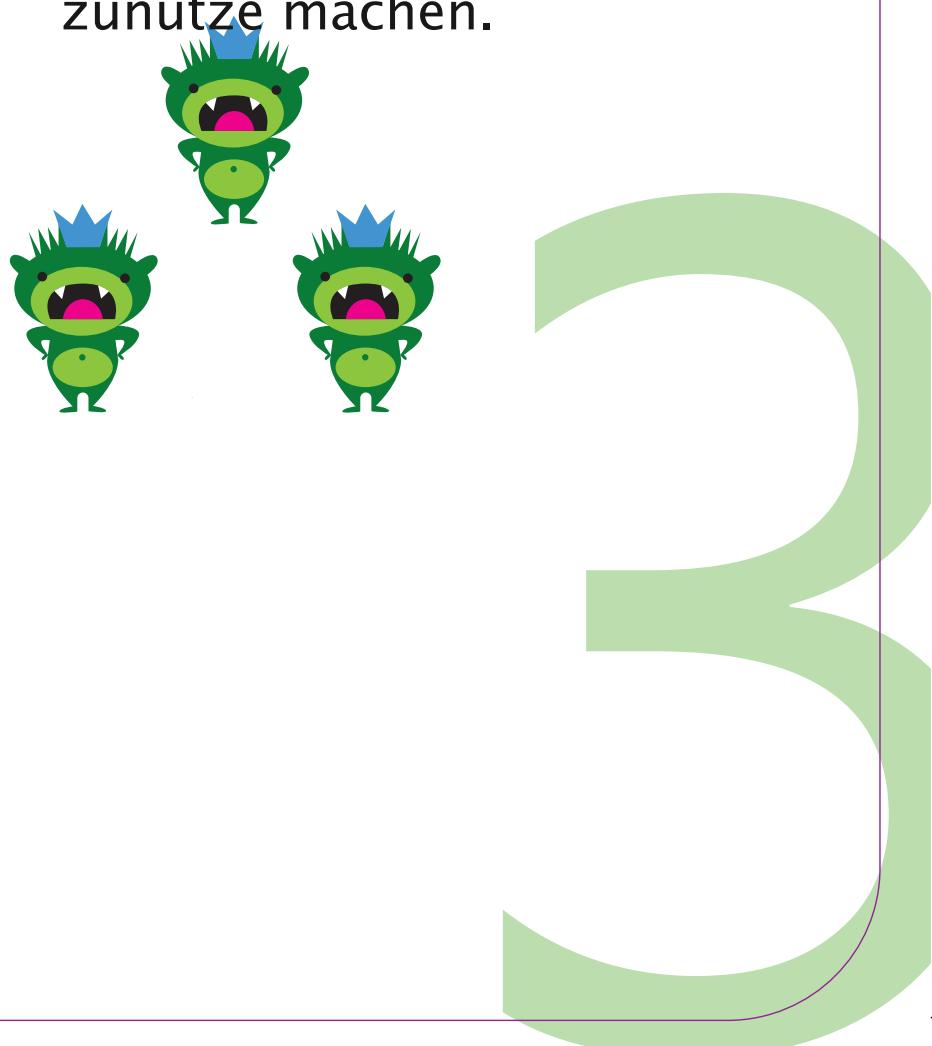
Sie haben einen neuen
Spoofing Angriff erfunden.



3

Tampering

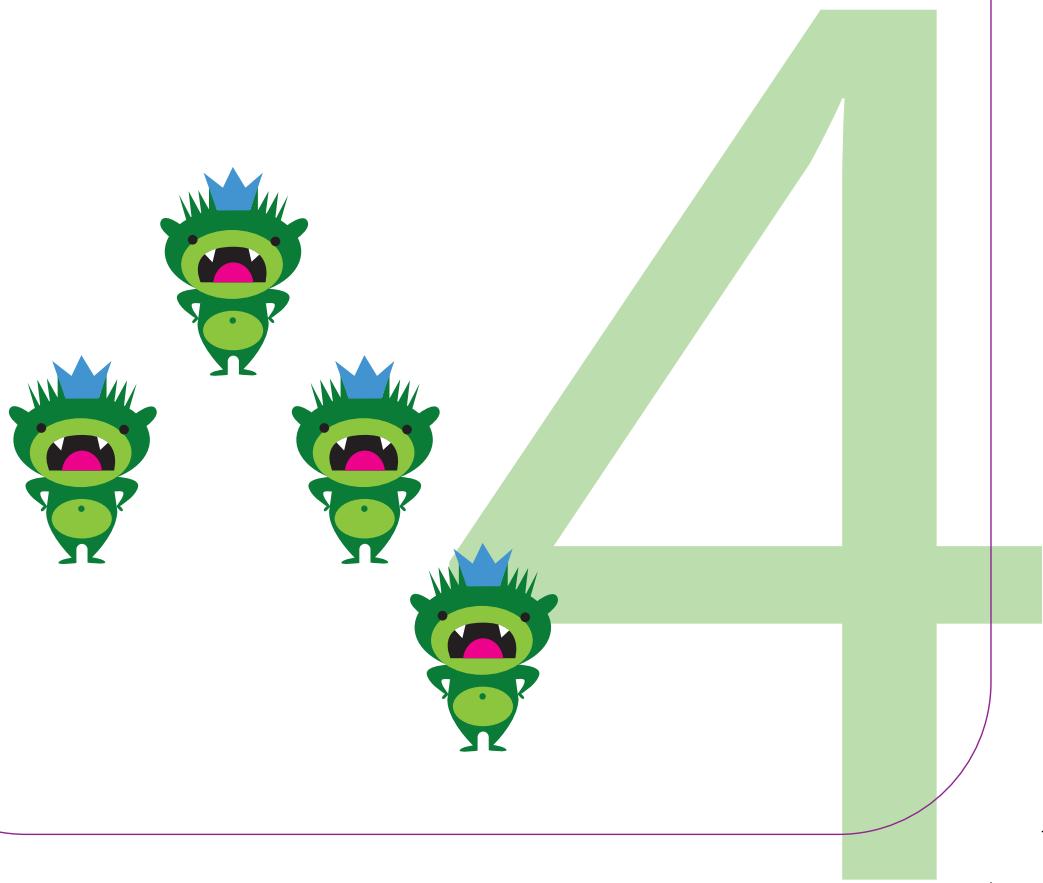
Statt auf Standard-Kryptografie zurück zu greifen, haben Sie sich selbst einen Mechanismus zur Gewährleistung von Integrität oder für den Schlüsselaustausch ausgedacht. Ein Angreifer kann sich das zunutze machen.



4

Tampering

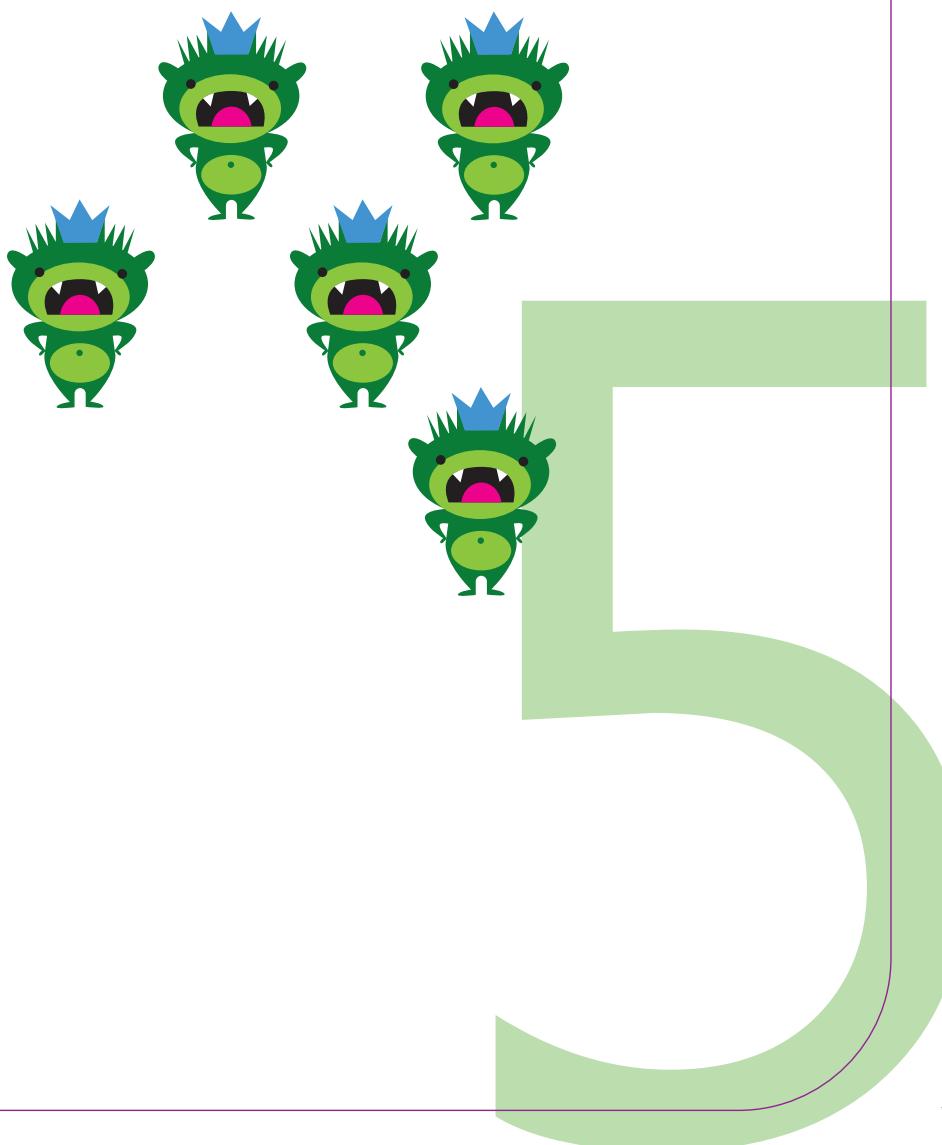
Ihr Code trifft Entscheidungen zur Zugangskontrolle an vielen unterschiedlichen Stellen, anstatt diese Funktion an zentraler Stelle (in einem Security Kernel) zu implementieren.



5

Tampering

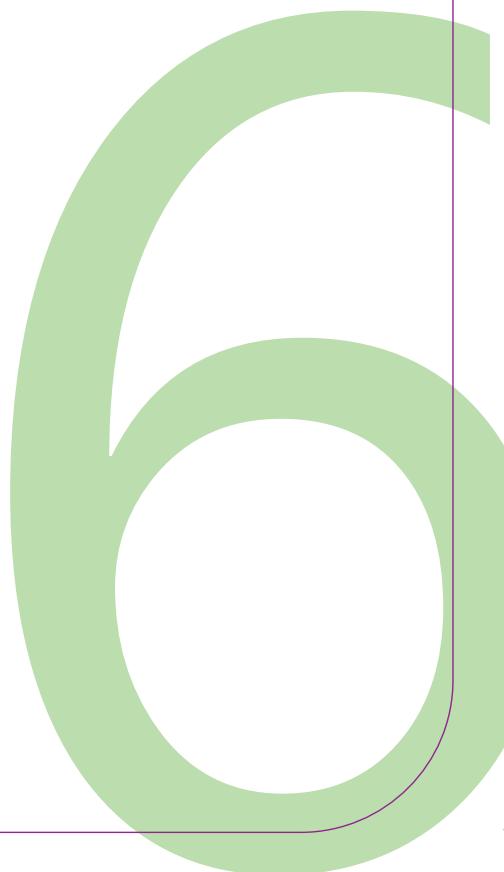
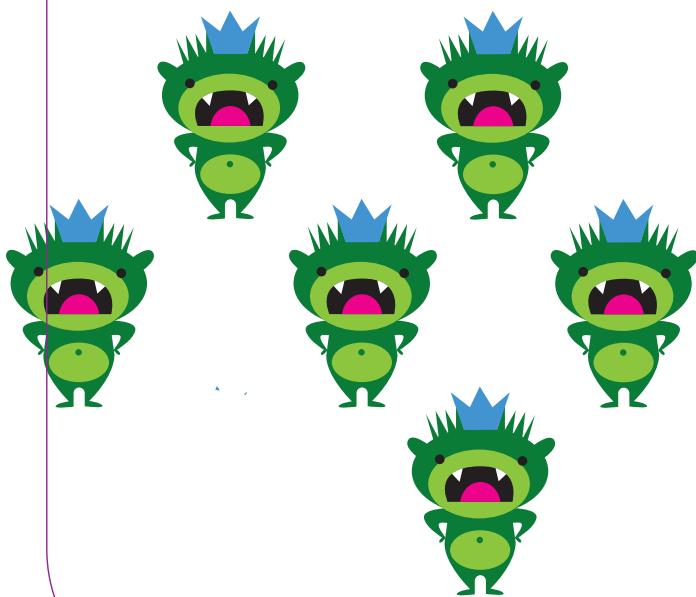
Ein Angreifer kann unbemerkt bereits übermittelte Daten erneut übertragen, weil Ihr Code keine Zeitstempel, Sequenznummern oder ähnliches nutzt, um dies zu verhindern oder zu erkennen.



6

Tampering

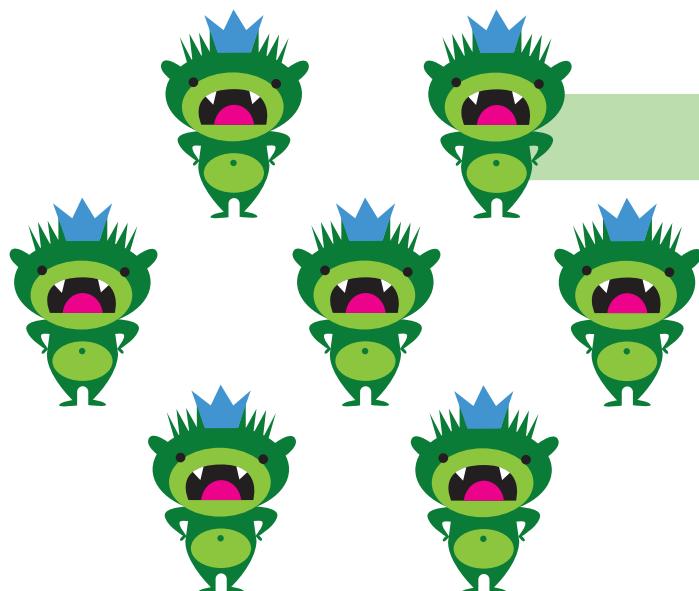
Ein Angreifer kann Daten an Speicherorten schreiben, an denen Ihr Code liegt oder die durch Ihren Code interpretiert werden.



7

Tampering

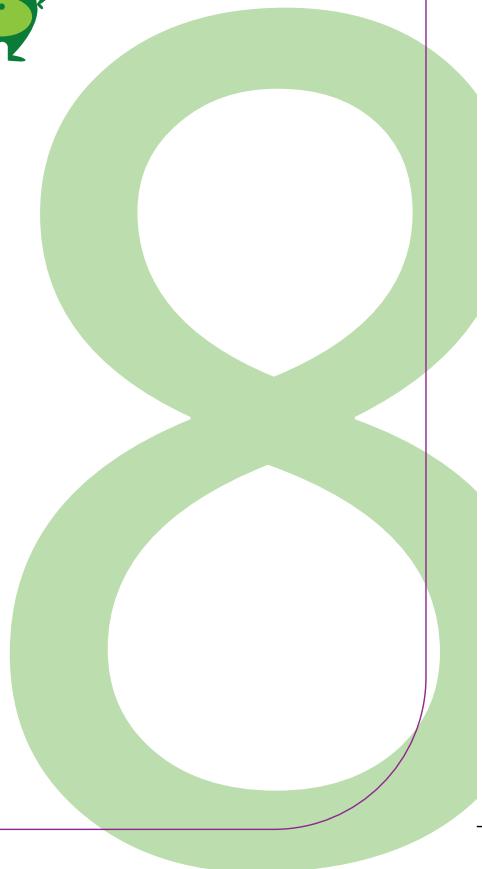
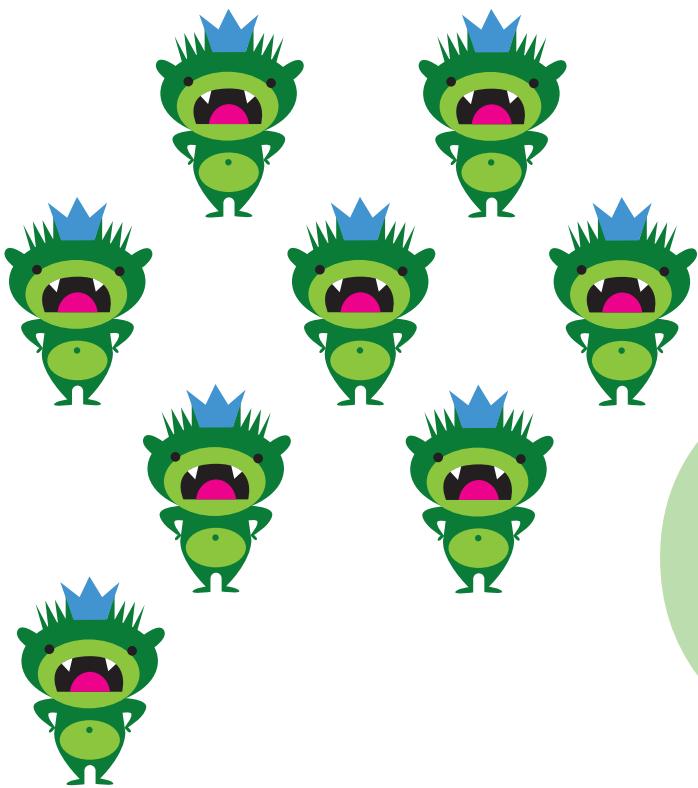
Ein Angreifer kann Berechtigungen umgehen, weil Sie Namen nicht kanonisieren (normalisieren), bevor Zugriffsrechte geprüft werden.



8

Tampering

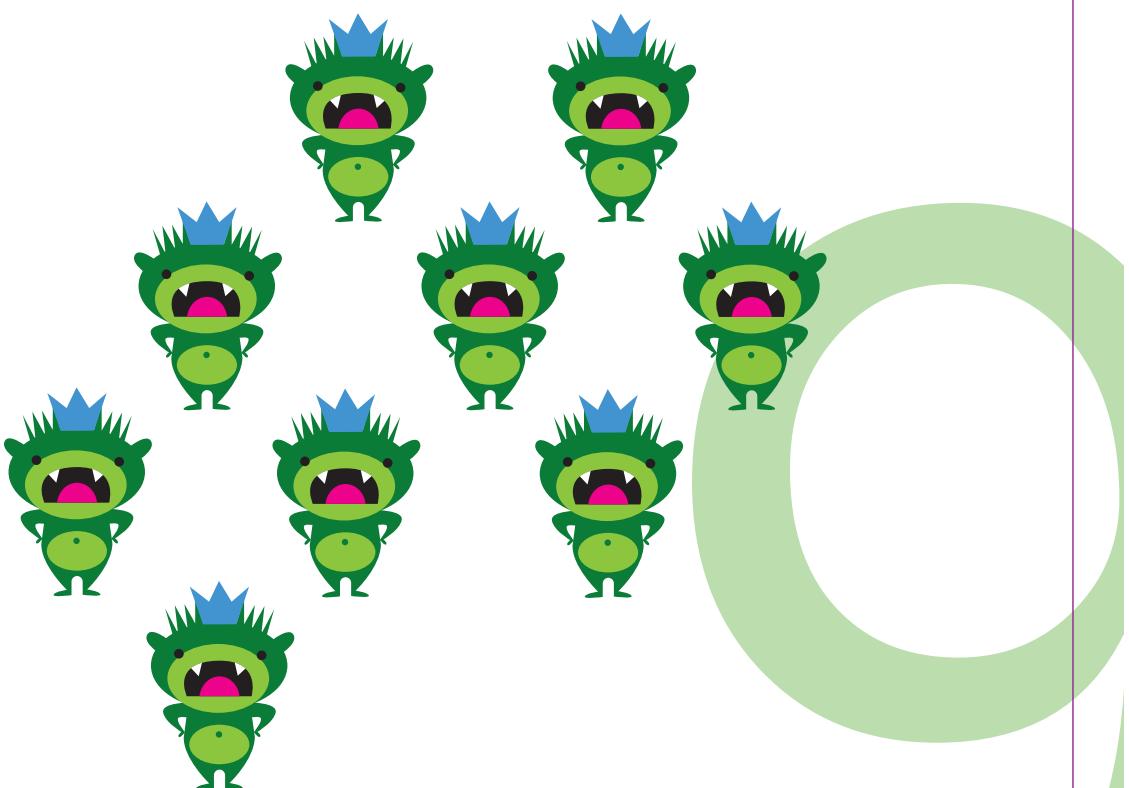
Ein Angreifer kann Daten manipulieren, die per Netzwerk übertragen werden, weil Ihr Code keine Integritätssicherung vorsieht.



9

Tampering

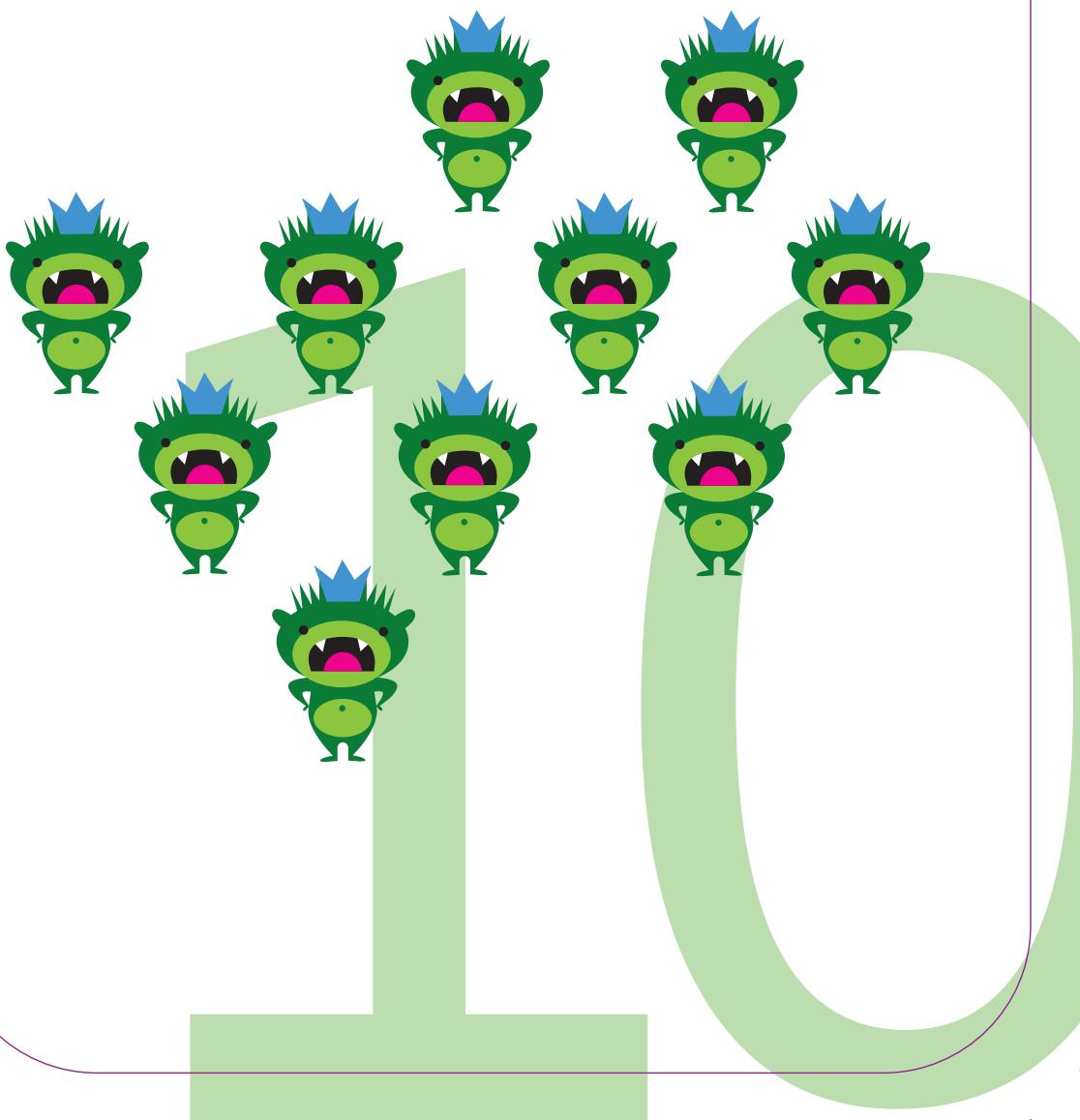
Ein Angreifer kann Statusinformationen beeinflussen.



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

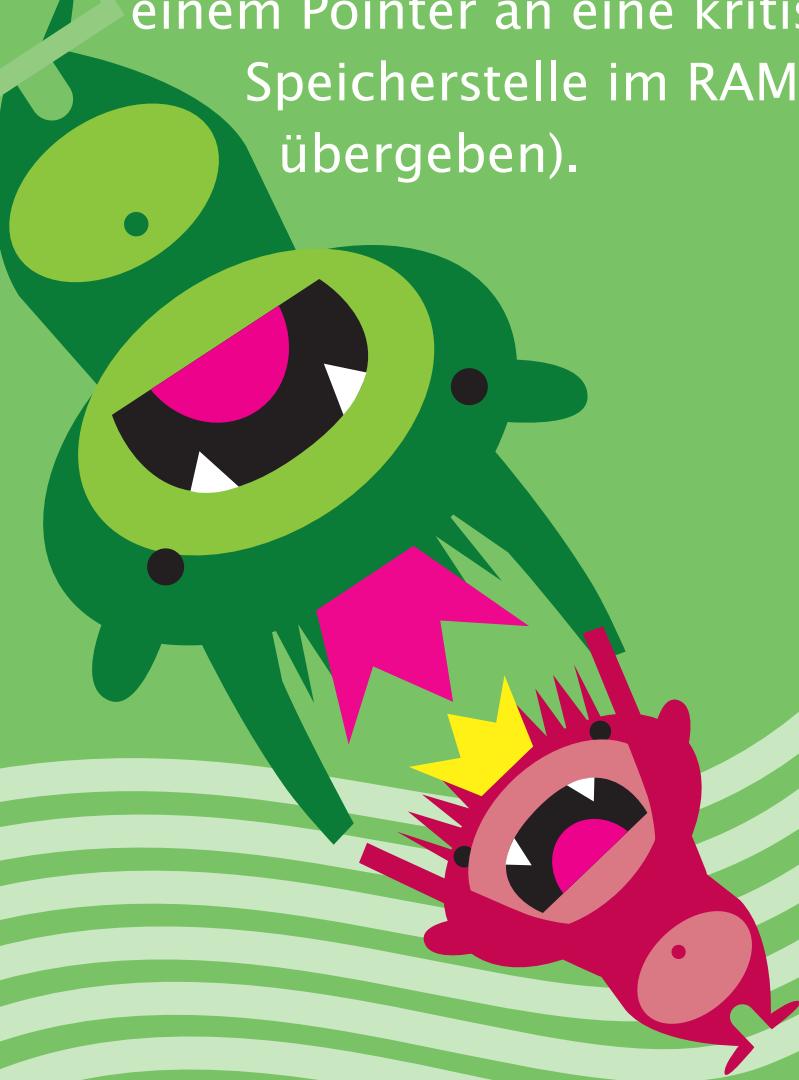
Ein Angreifer kann auf eine Ressource schreiben, weil es keine ACLs gibt, oder weil jeder berechtigt ist (world writable).



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

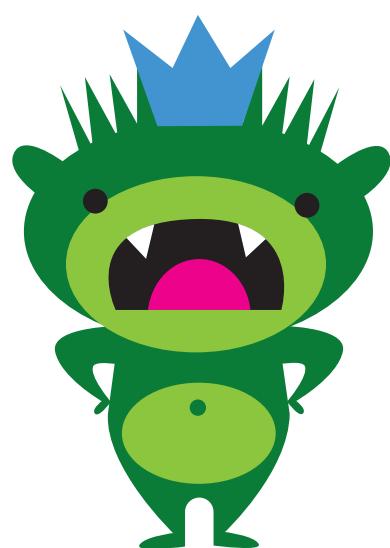
Ein Angreifer kann Code mithilfe eines Extension Points einbinden.



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.

R
R
R

2

3

Repudiation

Ein unprivilegierter 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 das Signierverfahren unsicher ist.



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

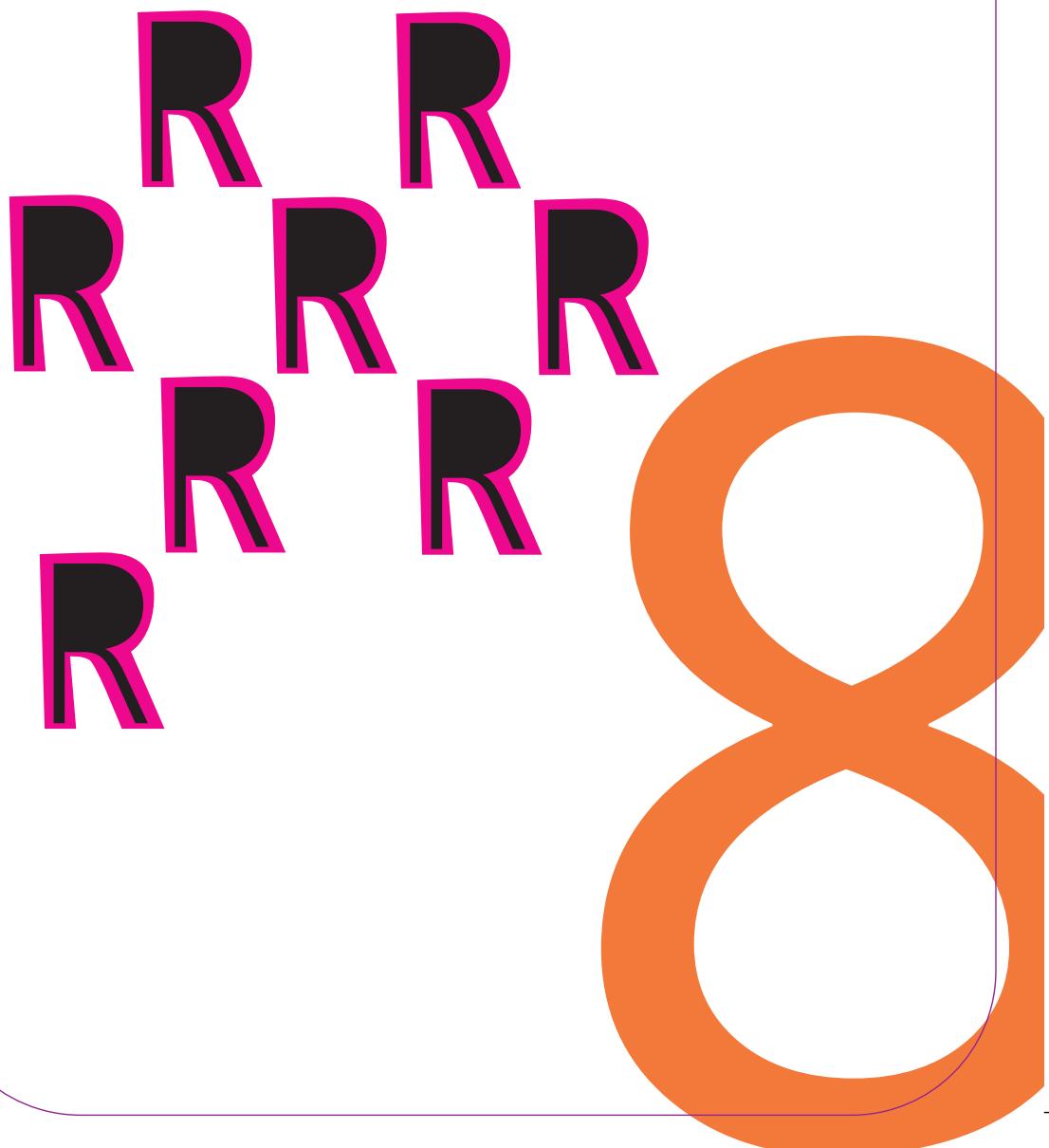
Ein Angreifer kann das Log zum Überlaufen bringen, so dass alte Logdaten überschrieben werden und somit verloren sind (wrap-around).



8

Repudiation

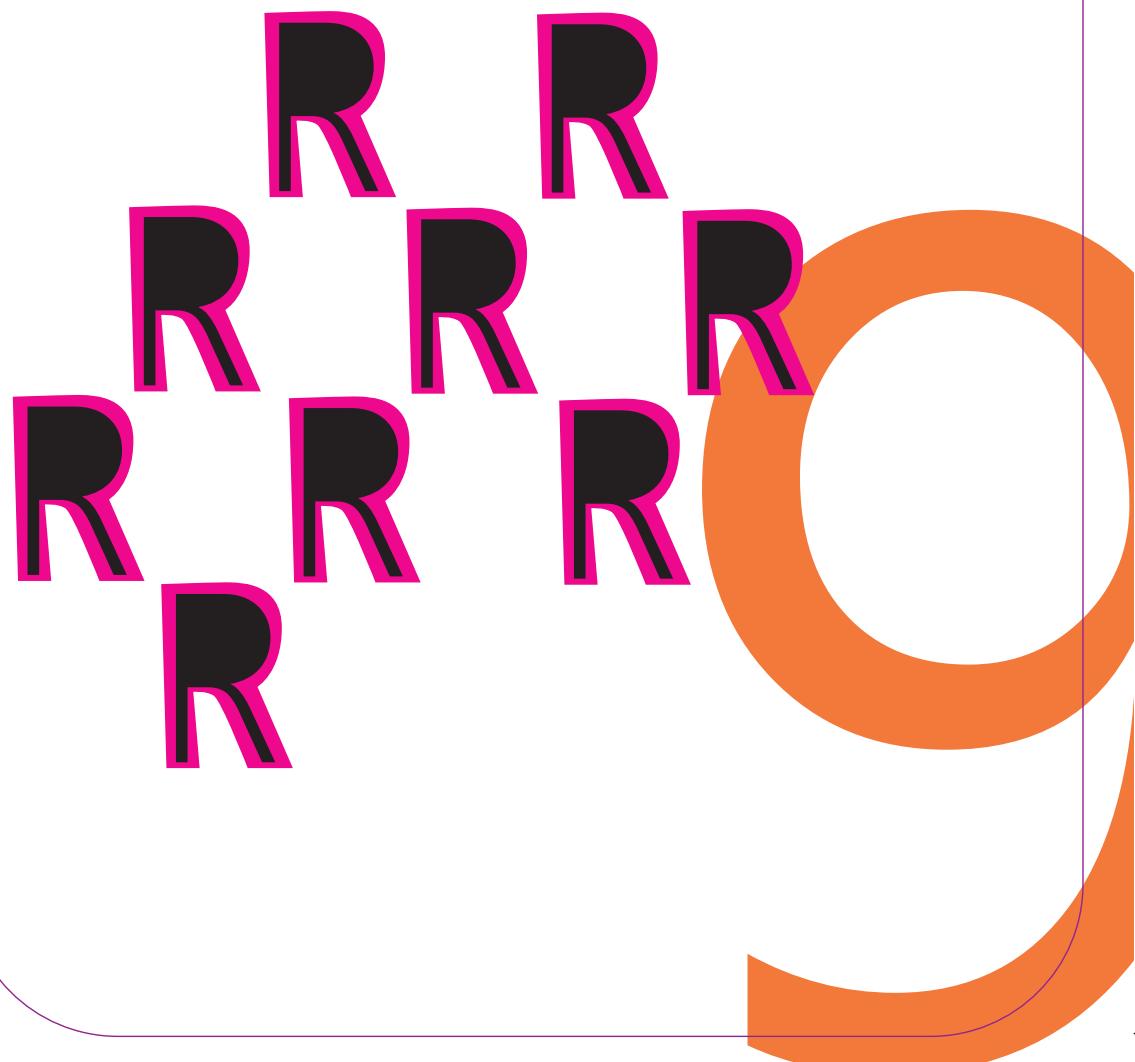
Ein Angreifer kann das Logging so austricksen, dass sicherheitsrelevante Logdaten nicht geschrieben werden oder durcheinander geraten.



9

Repudiation

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



10

Repudiation

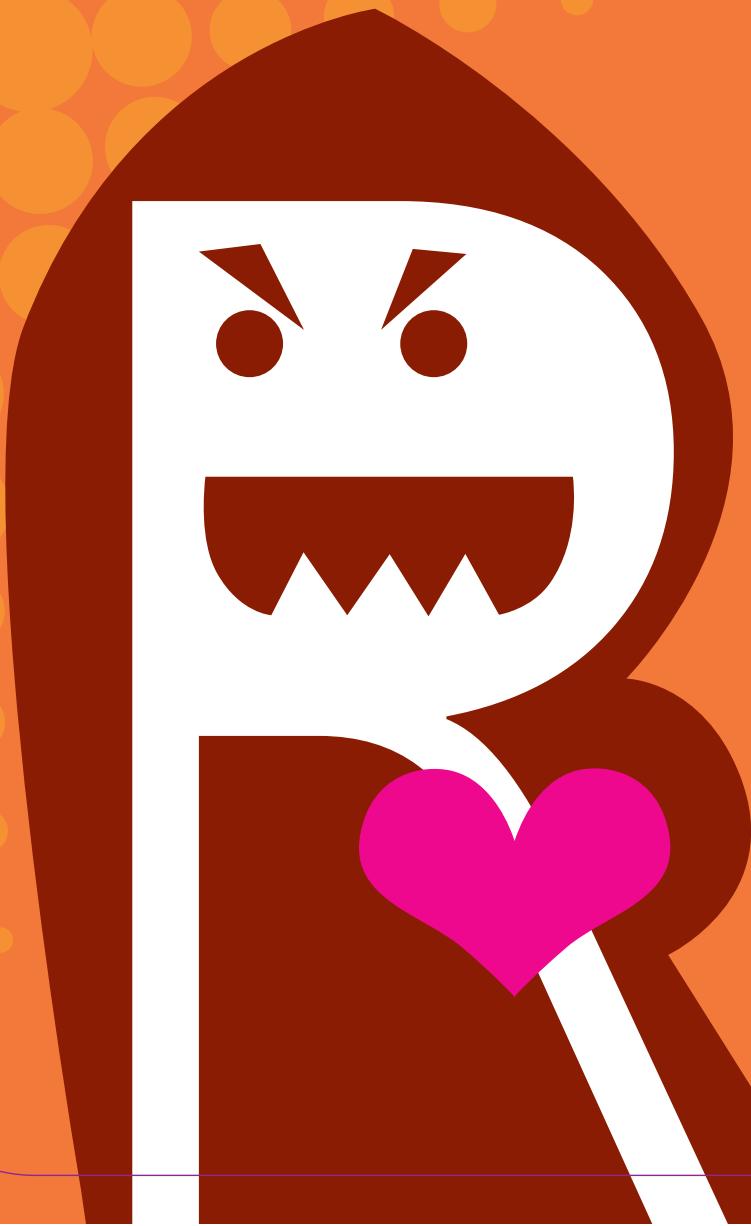
Ein Angreifer kann beliebige Logdaten in ein Logsystem einschleusen, weil die Logquellen nicht oder nur schwach authentisiert werden.



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.



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 sicherheits-relevante 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 kryptografischen 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.



Found it!

The illustration features a stylized white head with a smiling face, a green ponytail, and a green arrow pointing upwards. A yellow key is positioned near the bottom of the head. A white horse is running across a blue wave at the base of the head. A green speech bubble above the head contains the text "Found it!".

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 Netzwerkinformationen lesen, weil keine Kryptografie genutzt wird.



What!#@!
No cryptography was used?

A

Information Disclosure

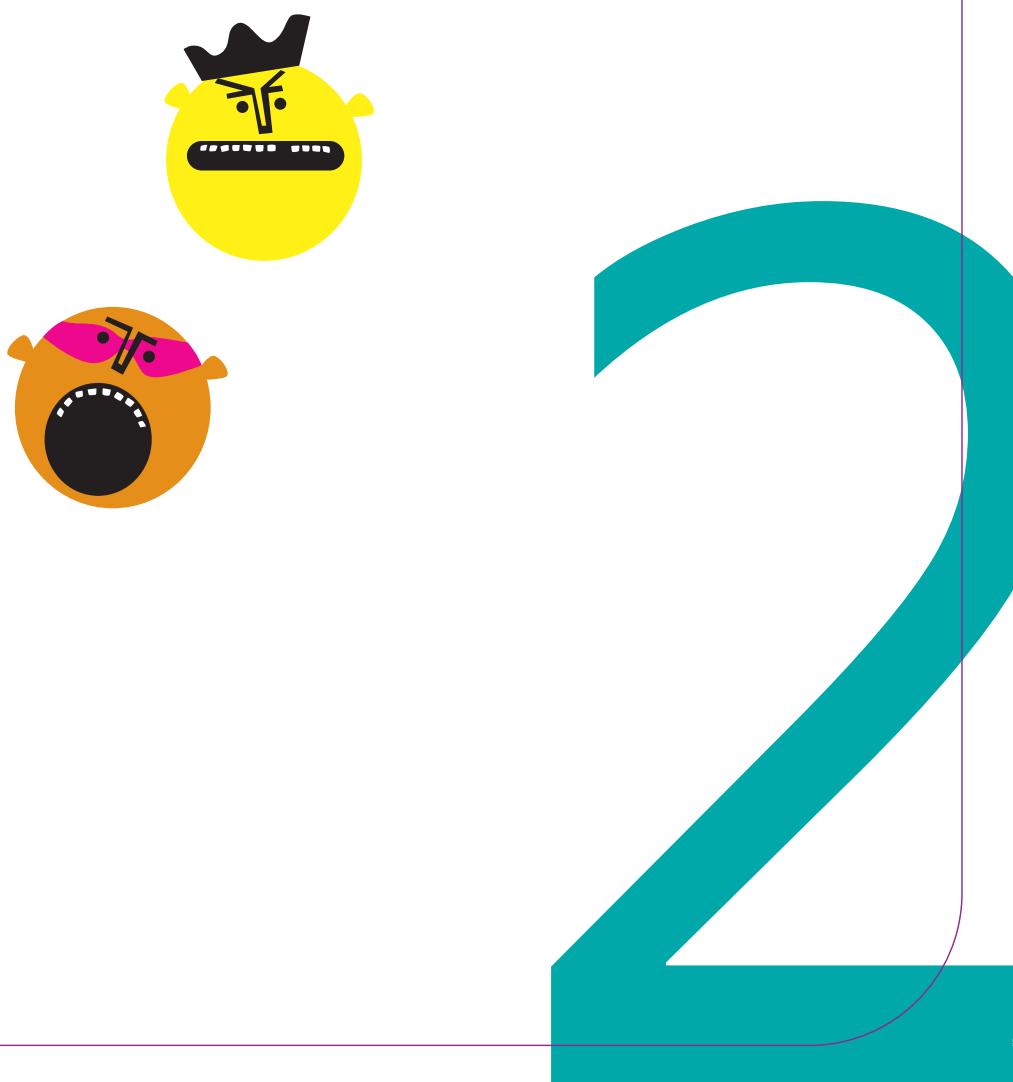
Sie haben einen neuen
Information Disclosure
Angriff erfunden.



2

Denial of Service

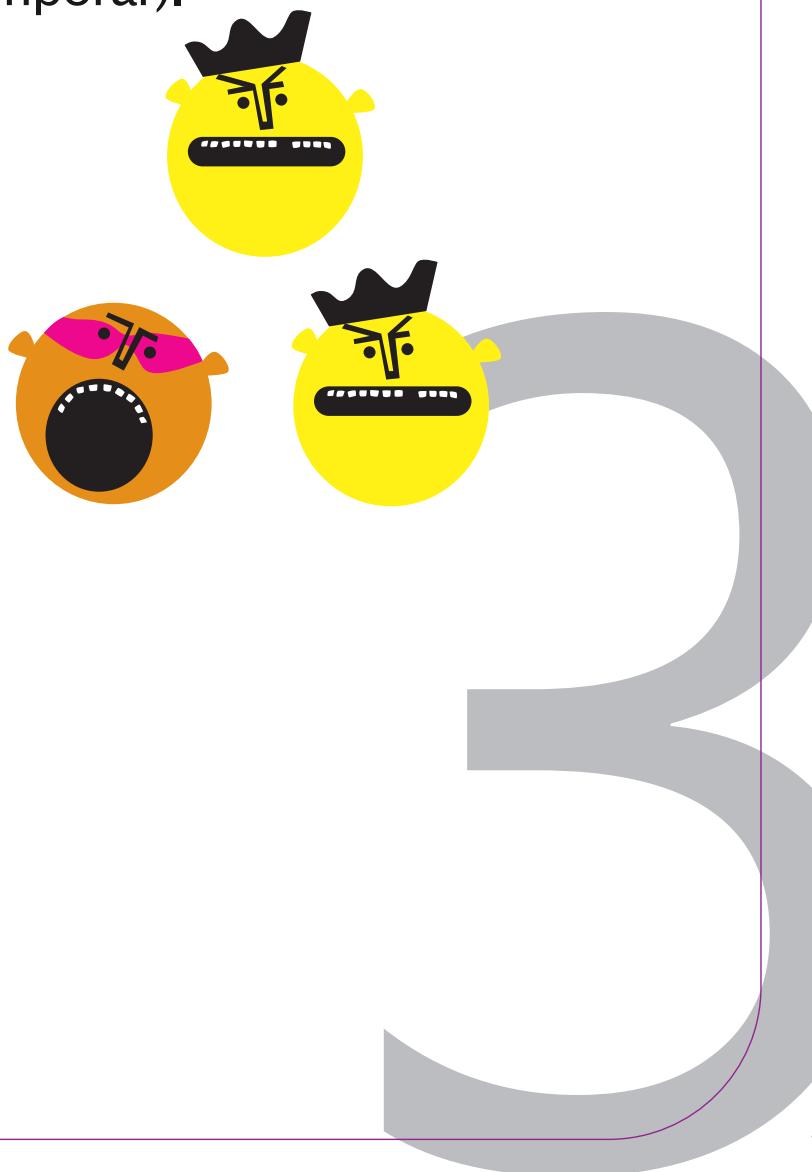
Ein Angreifer kann Ihr Authentisierungs-System unbrauchbar oder unverfügbar machen.



3

Denial of Service

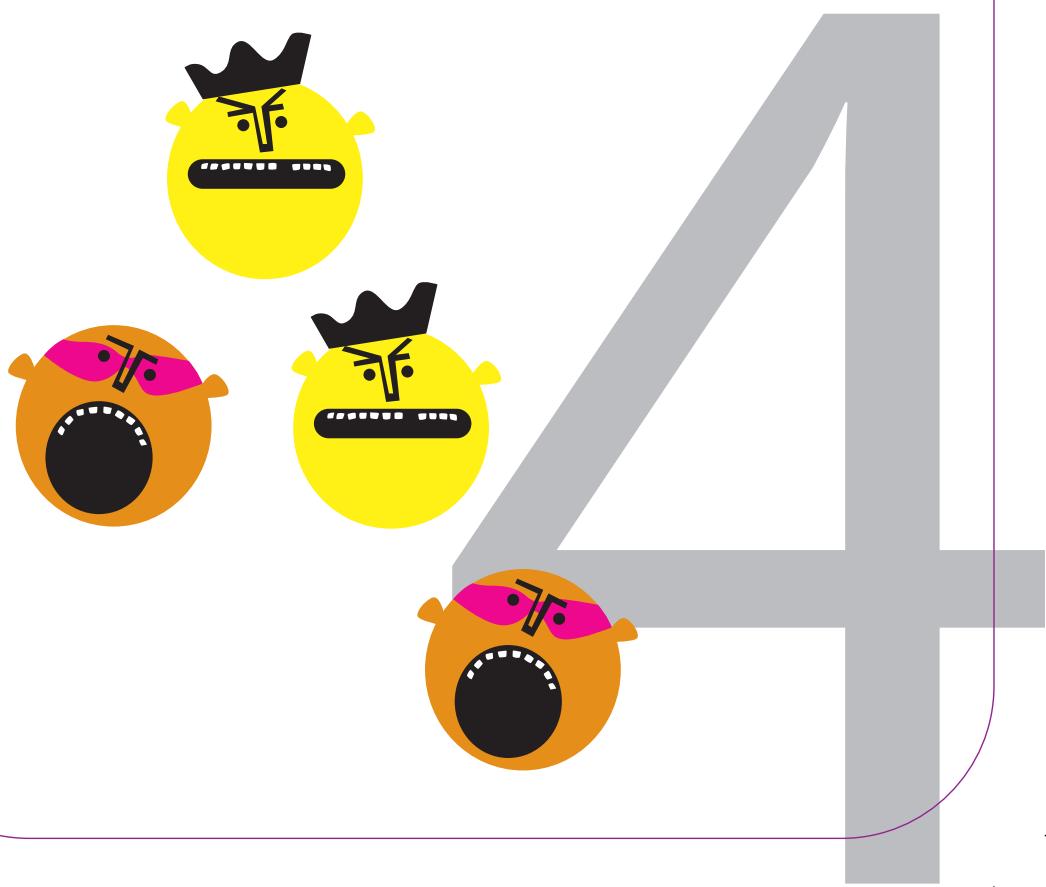
Ein Angreifer kann einen Client unverfügbar oder unbrauchbar machen, aber das Problem verschwindet, sobald der Angriff aufhört (Client, authentisiert, temporär).



4

Denial of Service

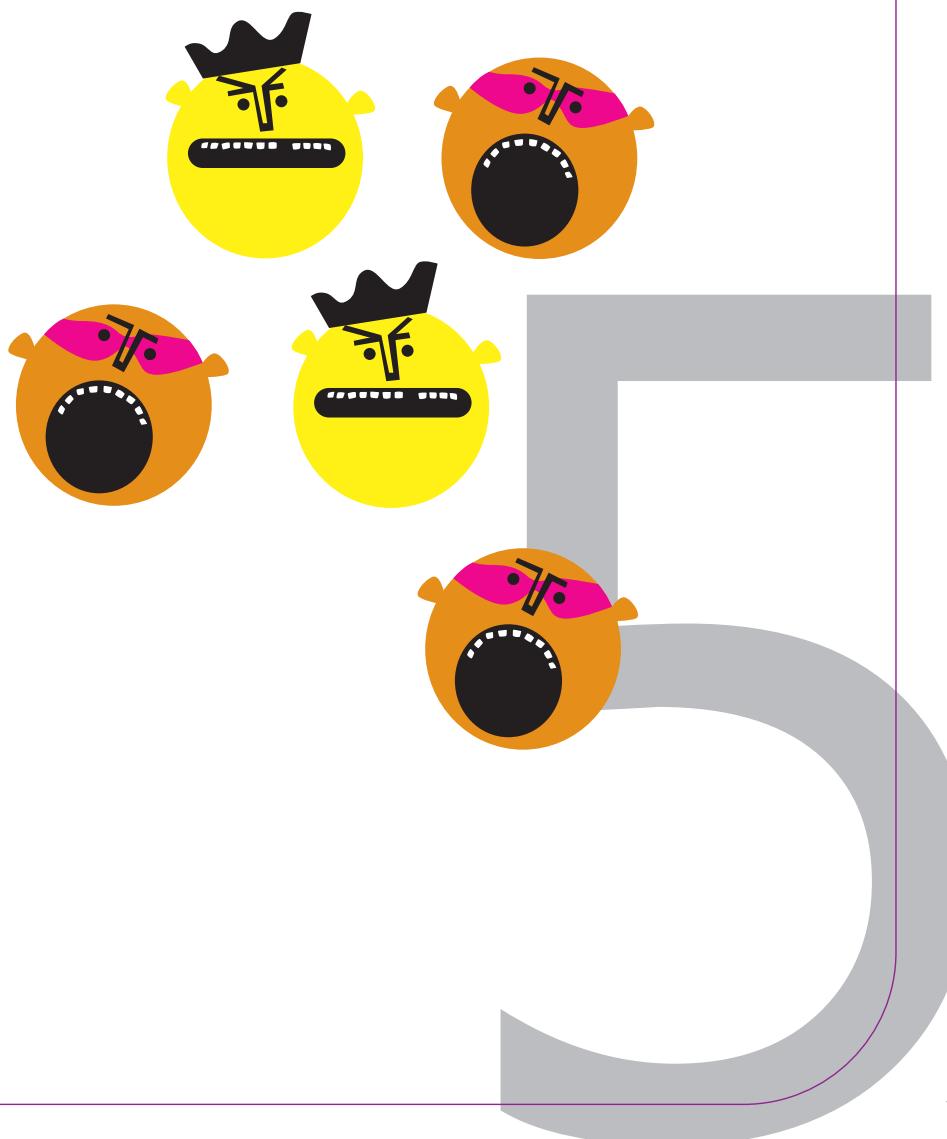
Ein Angreifer kann einen Server unverfügbar oder unbrauchbar machen, aber das Problem verschwindet, sobald der Angriff aufhört (Server, authentisiert, temporär).



5

Denial of Service

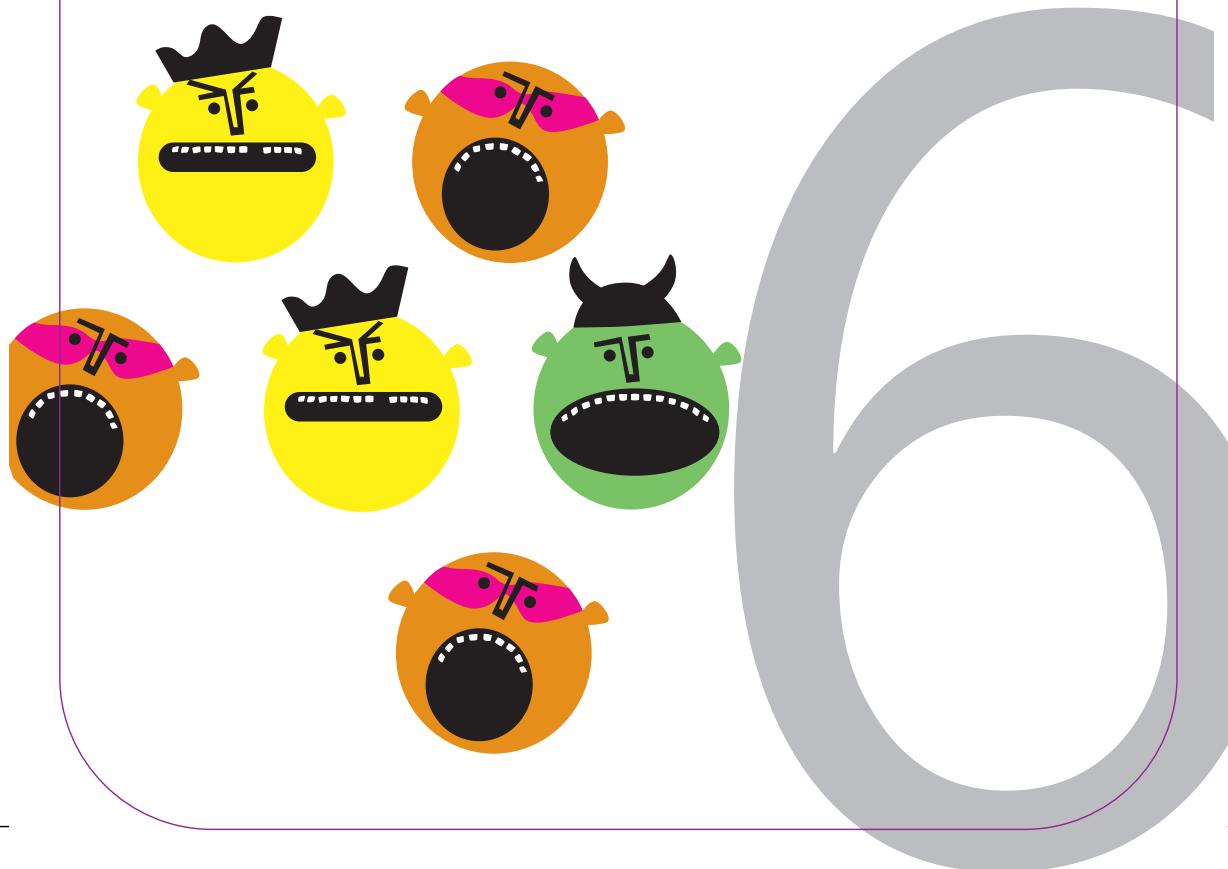
Ein Angreifer kann einen Client unverfügbar machen, ohne dass eine Authentisierung stattgefunden hat. Das Problem verschwindet nach dem Angriff (Client, anonym, temporär).



6

Denial of Service

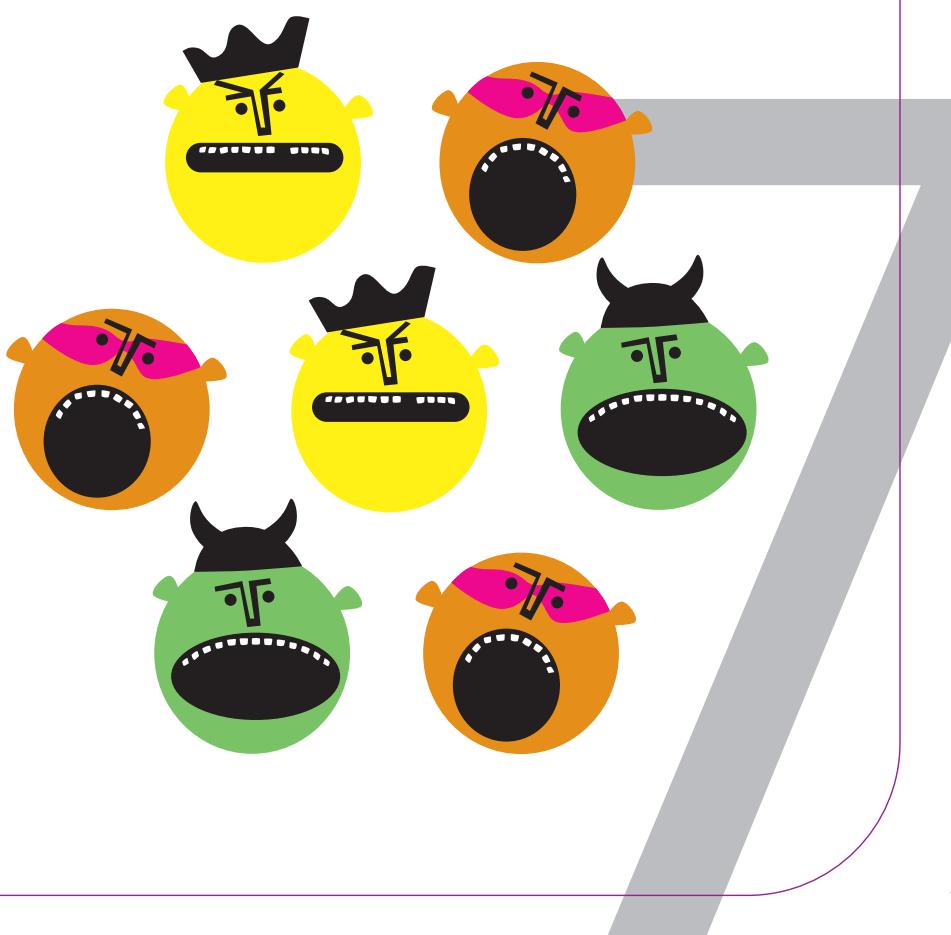
Ein Angreifer kann einen Server unverfügbar machen, ohne dass eine Authentisierung stattgefunden hat. Das Problem verschwindet nach dem Angriff (Server, anonym, temporär).



7

Denial of Service

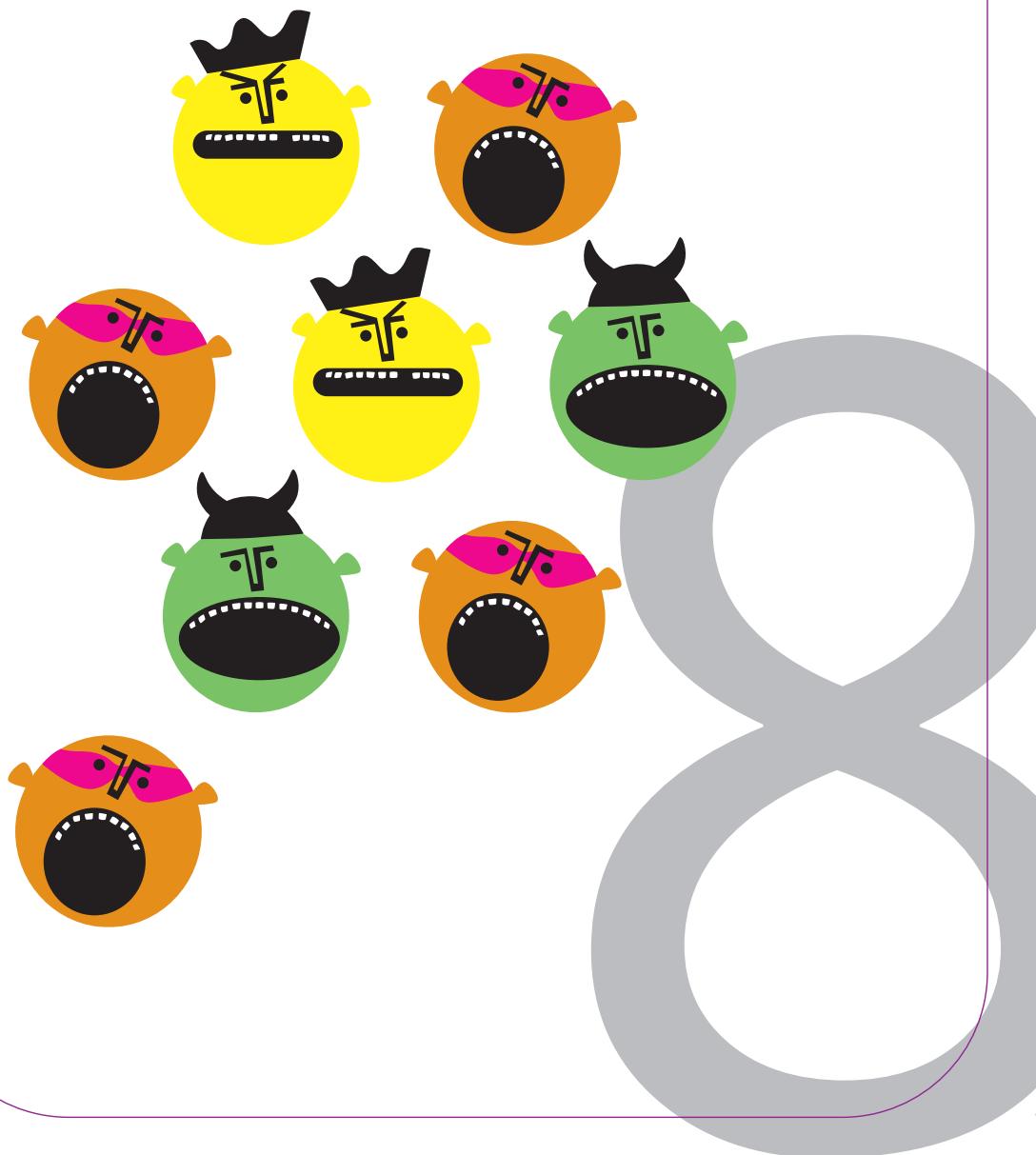
Ein Angreifer kann einen Client unverfügbar machen und das Problem besteht fort, nachdem der Angriff aufgehört hat (Client, authentisiert, persistent).



8

Denial of Service

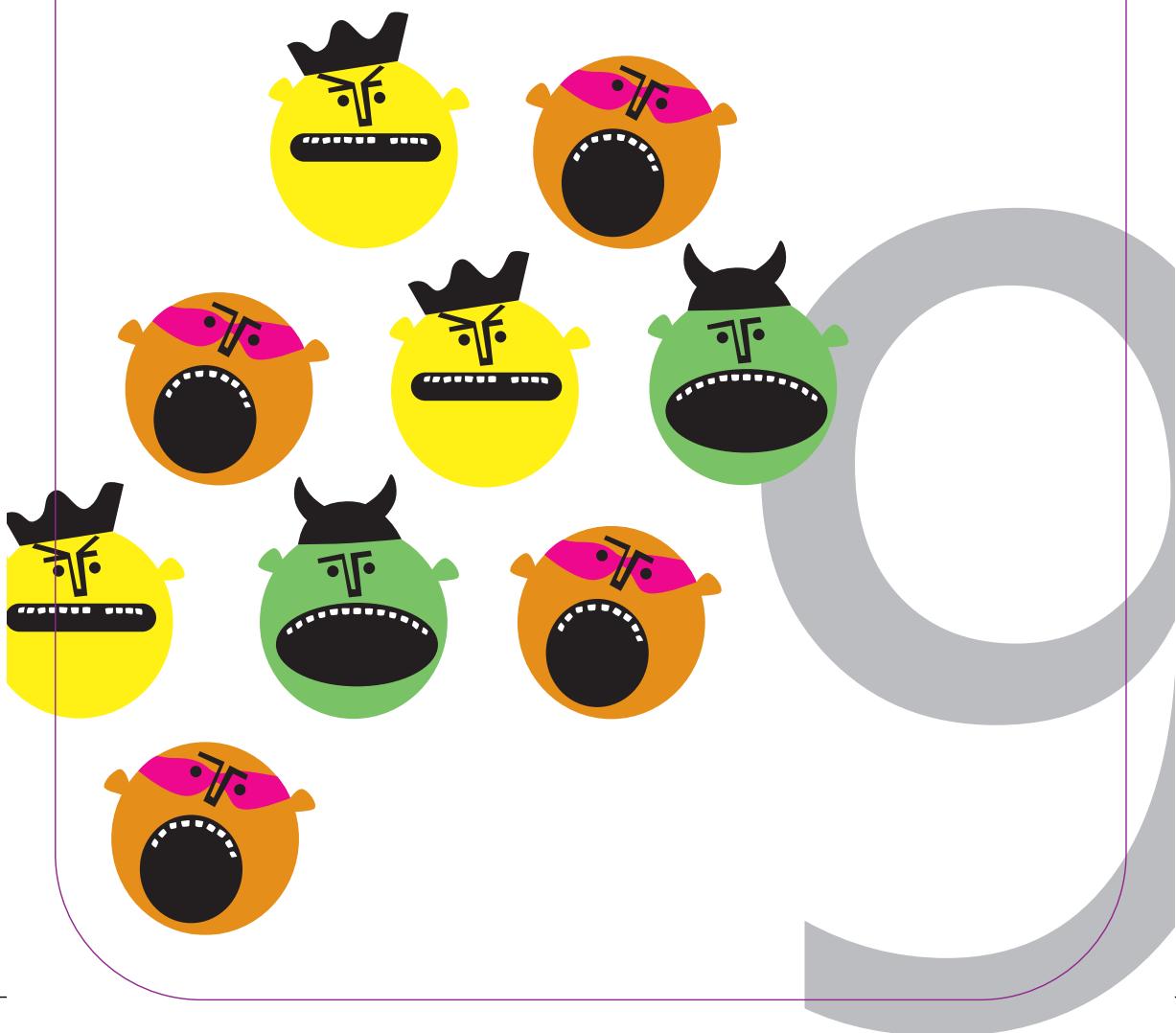
Ein Angreifer kann einen Server unverfügbar machen und das Problem besteht fort, nachdem der Angriff aufgehört hat (Server, authentisiert, persistent).



9

Denial of Service

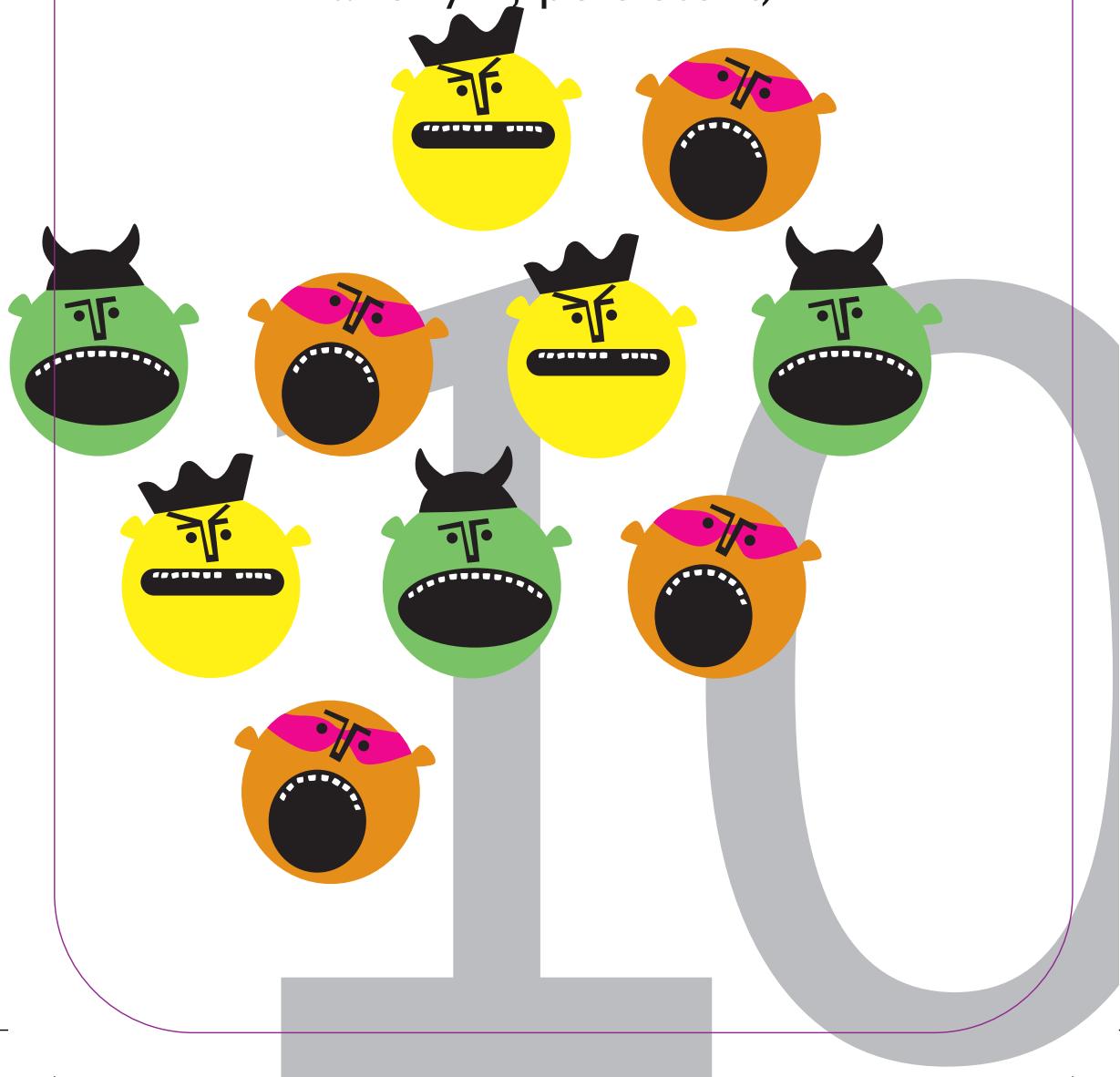
Ein Angreifer kann einen Client unverfügbar machen, ohne dass je eine Authentisierung stattgefunden hat, und das Problem besteht fort, nachdem der Angriff aufgehört hat (Client, anonym persistent).



10

Denial of Service

Ein Angreifer kann, ohne zu authentisieren, einen Server unverfügbar machen. Das Problem besteht fort, nachdem der Angriff aufgehört hat (Server, anonym, persistent).



J

Denial of Service

Ein Angreifer kann das Logging-Subsystem außer Betrieb setzen.



Q

Denial of Service

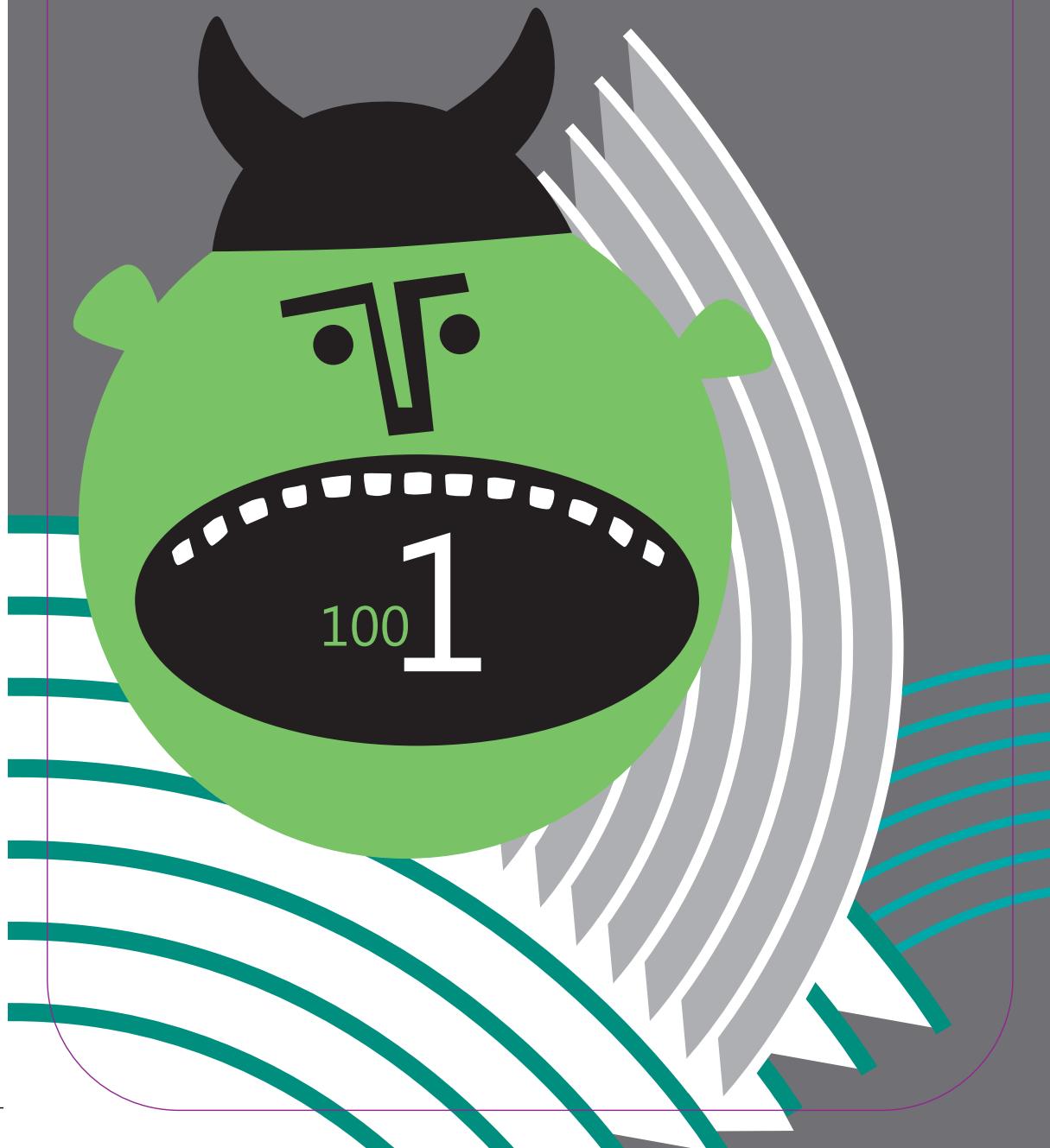
Ein Angreifer kann eine verwundbare Systemkomponente missbrauchen, um eine (volumenbasierte) DoS Attacke um den Faktor 10 zu verstärken.



K

Denial of Service

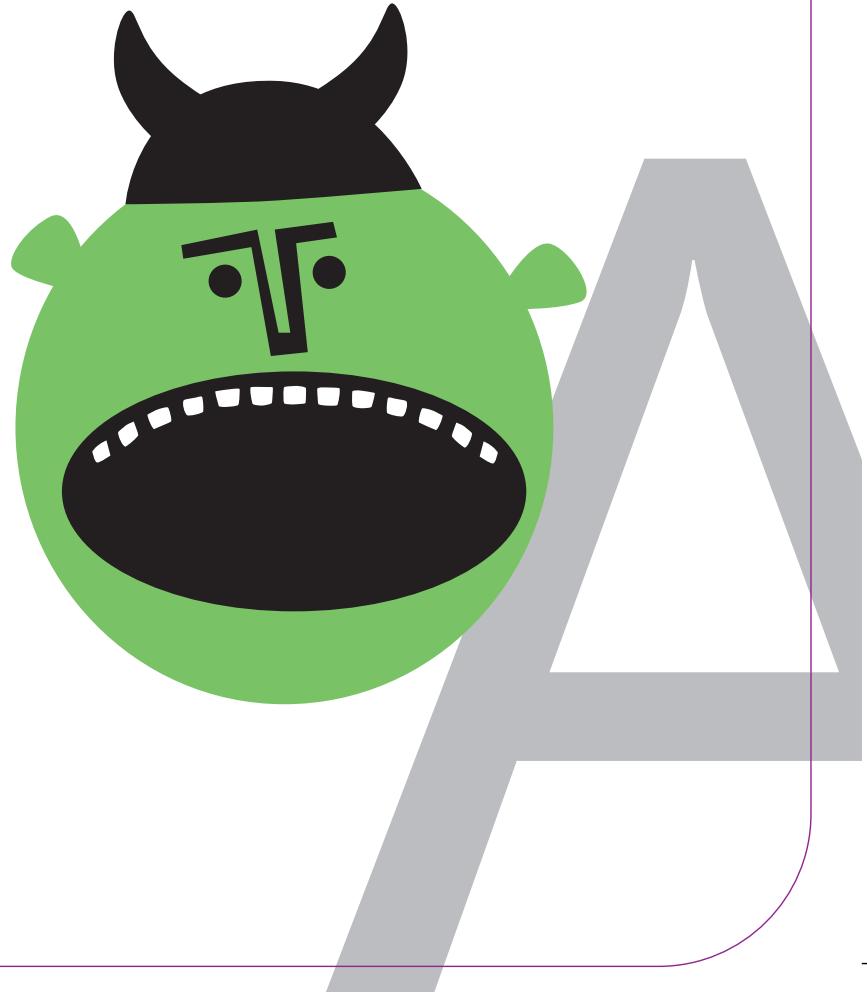
Ein Angreifer kann eine verwundbare Systemkomponente missbrauchen, um eine DoS Attacke mehr als 100fach zu verstärken.



A

Denial of Service

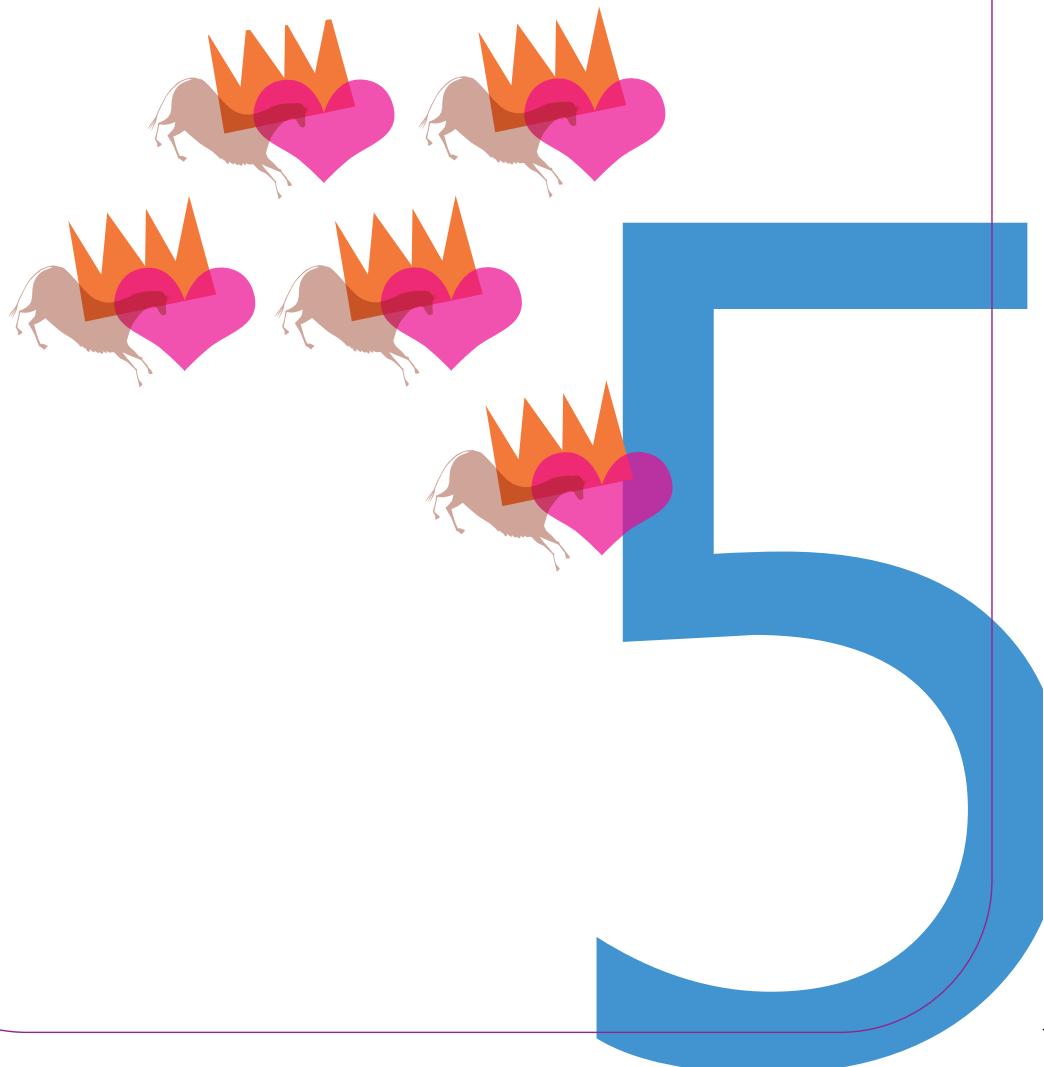
Sie haben eine neue Denial of Service Attacke erfunden.



5

Elevation of Privilege

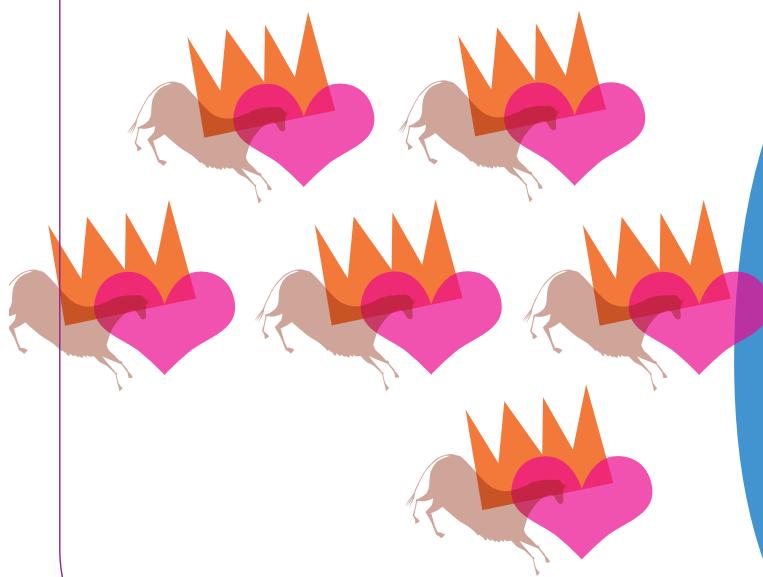
Ein Angreifer kann Einfluss darauf nehmen, welche Art Validierung Daten durchlaufen, die jeweils unterschiedliche Ergebnisse liefern.



6

Elevation of Privilege

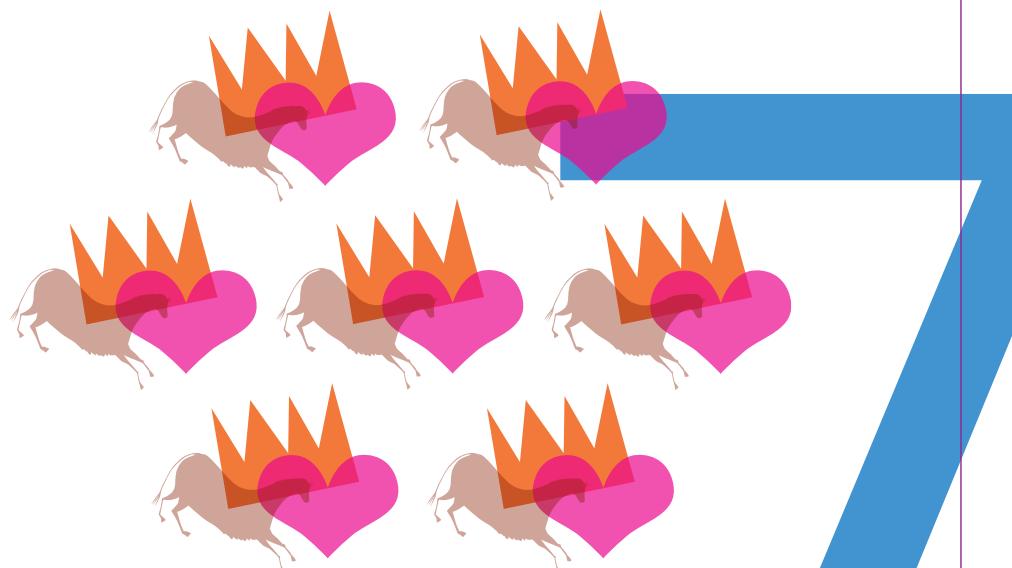
Ein Angreifer kann Berechtigungen, die Ihr Programm verlangt, aber nicht wirklich benötigt, für sich ausnutzen.



7

Elevation of Privilege

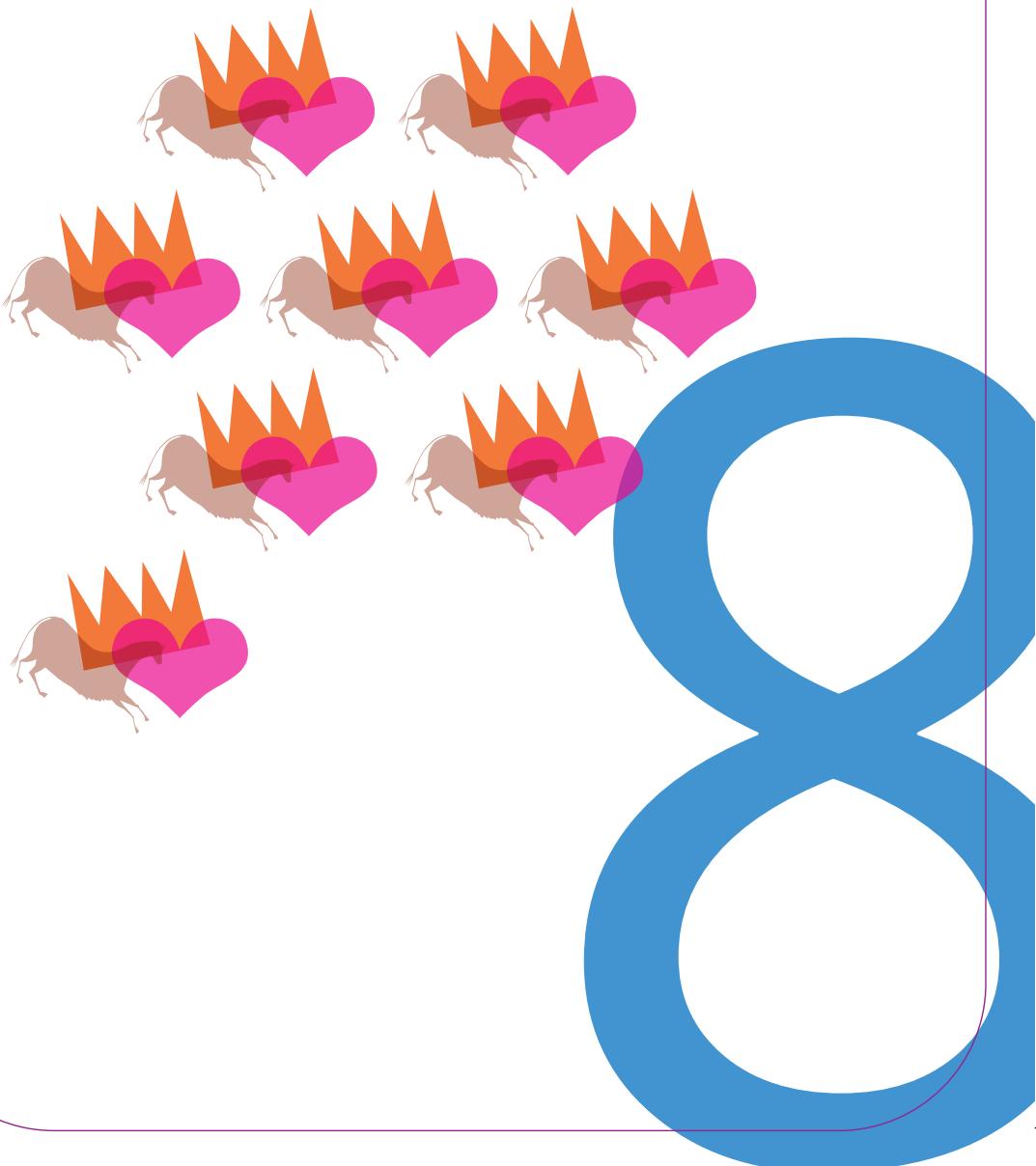
Ein Angreifer kann einen Pointer über eine Trust-Boundary hinweg angeben, anstatt Daten eingeben zu müssen, die eine Validierung durchlaufen.



8

Elevation of Privilege

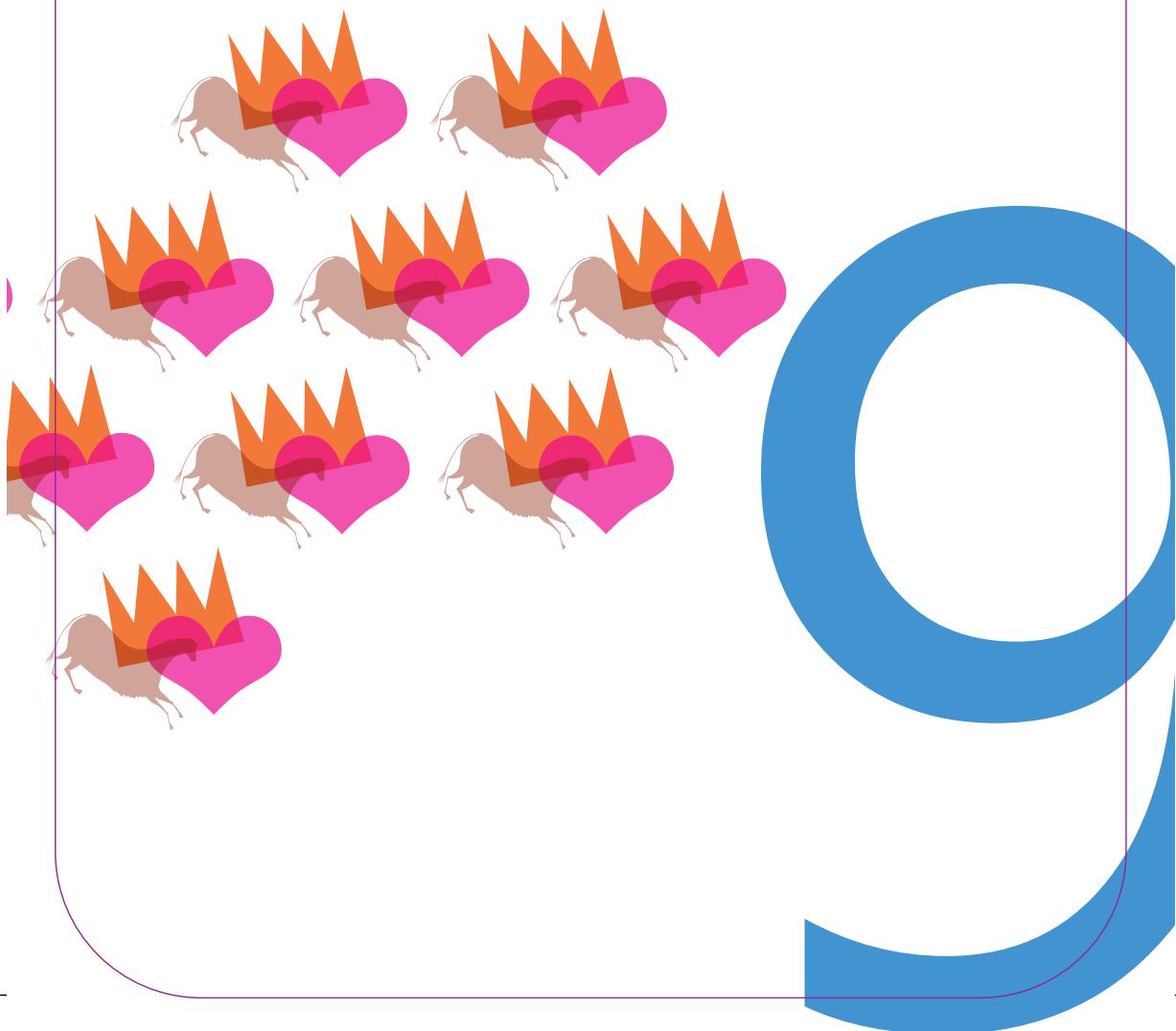
Eingegebene Daten befinden sich während der Überprüfung noch unter Kontrolle des Angreifers und werden später jenseits der Trust Boundary genutzt.



9

Elevation of Privilege

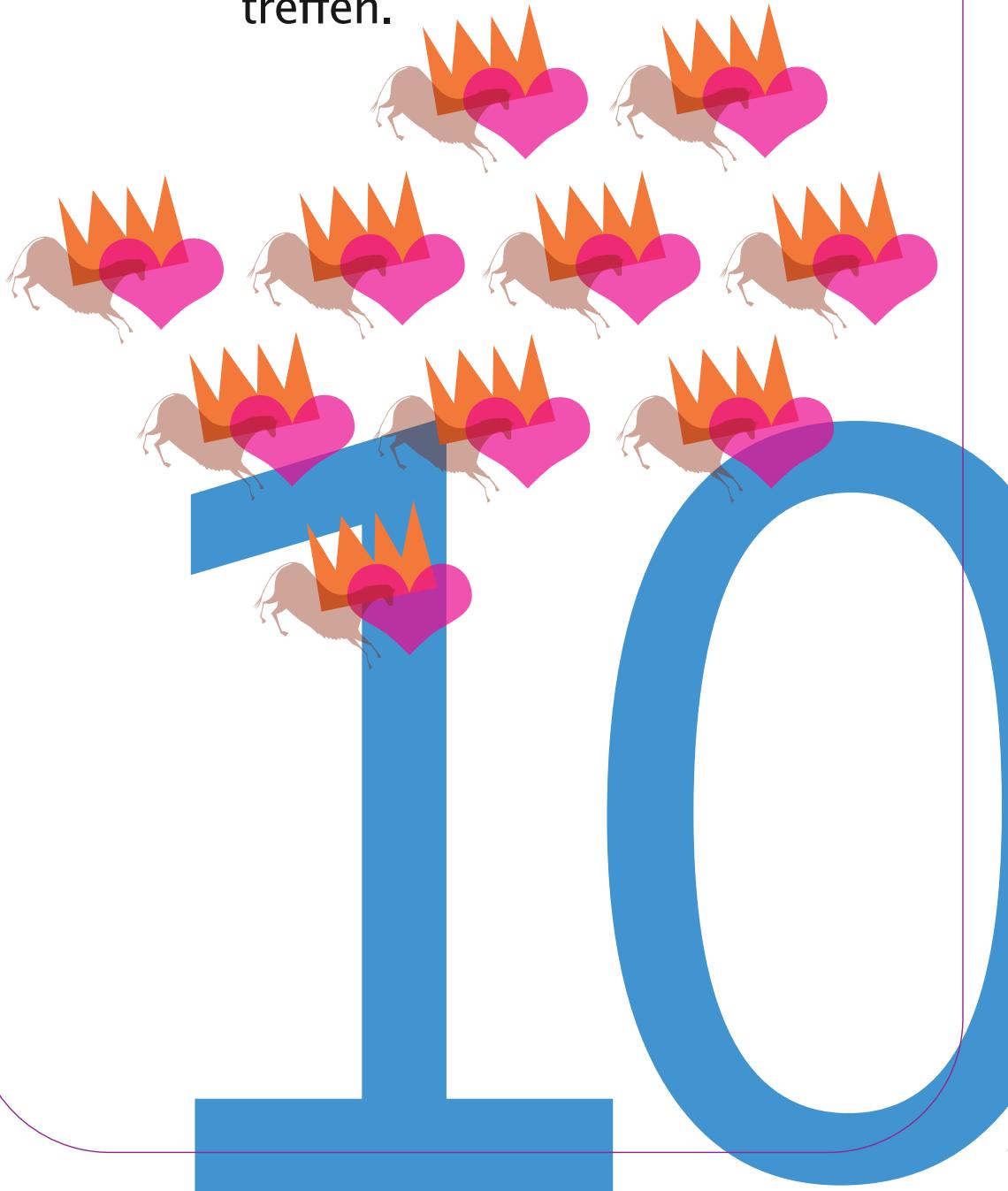
Caller (aufrufende Funktionen) haben keine Möglichkeit zu überprüfen, welche Validierung Ihr Programm auf die übergebenen Daten anwendet, bevor sie diese weitergeben.



10

Elevation of Privilege

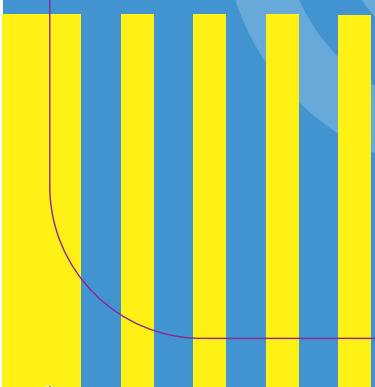
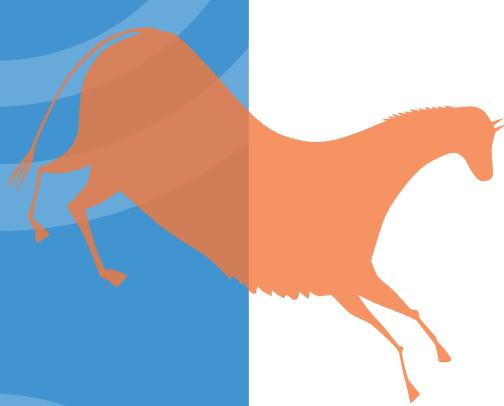
Es ist für einen Aufrufer/Caller nicht klar ersichtlich, welche Sicherheitsannahmen Sie treffen.



J

Elevation of Privilege

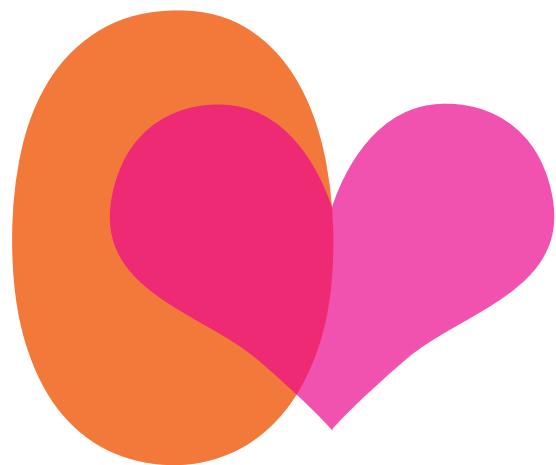
Ein Angreifer kann Eingaben zum Nutzer zurückspiegeln, z.B. per Cross-Site-Scripting (XSS).





Elevation

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





K

Elevation of Privilege

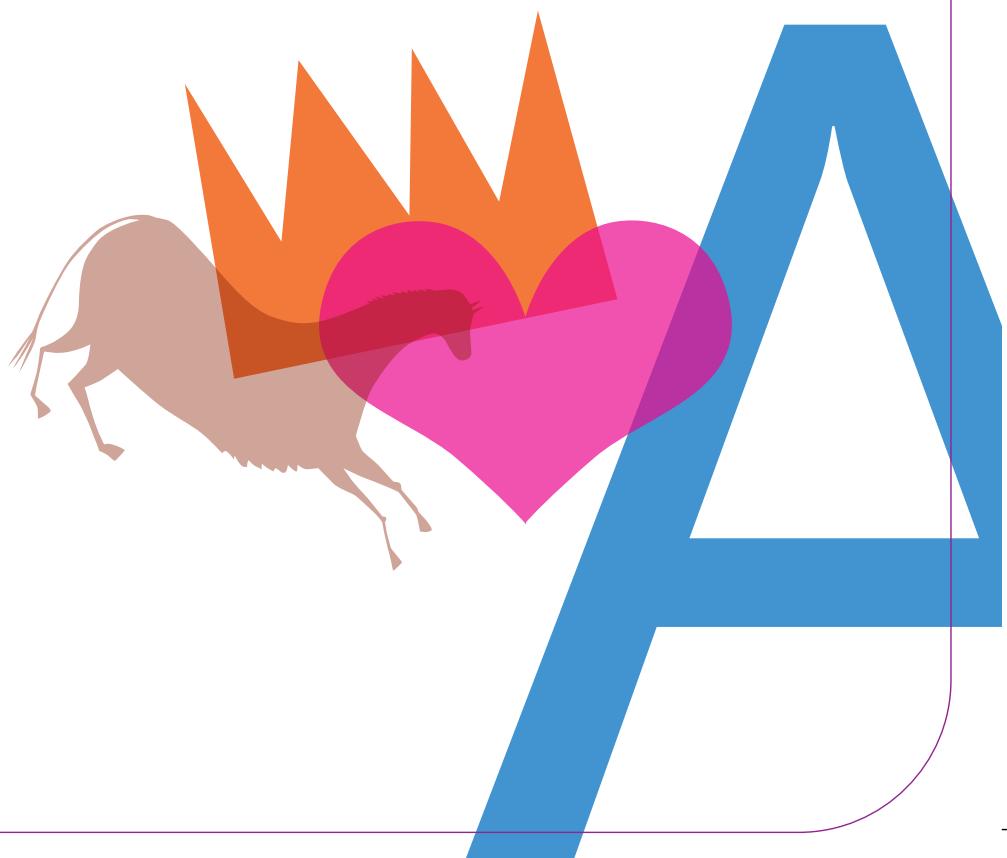
Ein Angreifer kann ein Kommando einschleusen, welches vom System mit einer höheren Berechtigung ausgeführt wird.



A

Elevation of Privilege

Sie haben einen neuen
Elevation of Privilege Angriff
erfunden.





Spoofing

2. Ein Angreifer "sitzt" (lauscht) auf einem zufälligen Port, der der Server üblicherweise nutzt.
3. Ein Angreifer kann alle mögl. Credentials der Reihe nach durchprobieren und es gibt keinen Mechanismus, der ihn ausbremsst (online oder offline).
4. Ein Angreifer kann sich anonym verbinden, weil Sie davon ausgehen, dass Authentisierung auf einer höheren Schicht stattfindet.
5. Ein Angreifer kann einen Client verwirren, weil es zu viele Wege gibt, einen Server zu identifizieren.
6. 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. Ein Angreifer kann sich zu einem Server oder Peer über einen nicht authentisierten unverschlüsselten Kanalverbinden.
8. Ein Angreifer kann auf einem Server gespeicherte Credentials stehlen und wieder verwenden (z.B. Schlüssel in einer für andere lesbaren Datei).
9. Ein Angreifer, der Zugang zu einem Passwort bekommt, kann es wieder verwenden (nutzen Sie stärkere Authentisierungsmethoden).

Fortsetzung siehe Rückseite

Spoofing



Tampering

3. 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 das zunutze machen.
4. Ihr Code trifft Entscheidungen zur Zugangskontrolle an vielen unterschiedlichen Stellen, anstatt diese Funktion an zentraler stelle zu implementieren
5. Ein Angreifer kann unbemerkt bereits übermittelte Daten erneut übertragen, weil Ihr Code keine Zeitstempel, Sequenznummern o.ä. nutzt, um dies zu verhindern oder zu erkennen.
6. Ein Angreifer kann Daten an Speicherorten schreiben, an denen Ihr Code liegt oder die durch Ihren Code interpretiert werden.
7. Ein Angreifer kann Berechtigungen umgehen, weil Sie Namen nicht kanonisieren (normalisieren), bevor Zugriffsrechte geprüft werden.
8. Ein Angreifer kann Daten manipulieren, die per Netzwerk übertragen werden, weil Ihr Code keine Integritätssicherung vorsieht.
9. Ein Angreifer kann Statusinformationen beeinflussen.
10. 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.

Fortsetzung siehe Rückseite

Tampering

R

Repudiation

2. Ein Angreifer kann den Inhalt von Logdaten beeinflussen, einen Log Reader darüber angreifen und es ist nicht dokumentiert, ob/wie verschiedene Logdaten validiert werden.
3. Ein unprivilegierter Nutzer oder Angreifer hat lesend Zugang zu interessanten Sicherheitsinformationen in den Logs.
4. Ein Angreifer kann digitale Signaturen manipulieren, weil Sie einen MAC Algorithmus statt einem Signierverfahren nutzen, oder weil das Signierverfahren unsicher ist.
5. Ein Angreifer kann Lognachrichten verändern, die übers Netz übertragen werden, weil kein starker Mechanismus zur Gewährleistung der Integrität implementiert ist.
6. Ein Angreifer kann einen Logeintrag ohne Zeitstempel erzeugen (oder die Logs haben generell keine Zeitstempel).
7. Ein Angreifer kann das Log zum Überlaufen bringen, so dass alte Logdaten überschrieben werden und somit verloren sind (wrap-around).
8. Ein Angreifer kann das Logging so austricksen, dass sicherheitsrelevante Logdaten nicht geschrieben werden oder durcheinander geraten.
9. Ein Angreifer kann einen Shared key nutzen, um sich als jemand anders auszugeben, so dass seine Aktionen ebenfalls unter dieser Identität mitgeloggt werden.

Fortsetzung siehe Rückseite

Repudiation



Information Disclosure

2. Ein Angreifer kann verschlüsselte Dateien mittels Brute-Force entschlüsseln, weil keine Sicherheitsmaßnahmen dagegen vorhanden sind.
3. Ein Angreifer kann sicherheitsrelevante Fehlermeldungen sehen
4. Ein Angreifer kann Dateninhalte lesen, weil die Nachrichten (z.B. E-Mails oder Cookies) nicht verschlüsselt sind, selbst wenn der Transportkanal verschlüsselt ist.
5. Ein Angreifer kann unter Umständen Daten lesen, die mit einem nicht standardisierten kryptografischem Algorithmus verschlüsselt sind.
6. 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. Ein Angreifer kann als "Man in the Middle" verschlüsselte Daten lesen, weil die Endpunkte einer Netzwerkverbindung nicht authentisiert sind.
8. Ein Angreifer kann (sensible) Informationen mithilfe eines Such-Indexers, Loggers oder eines anderen Mechanismus zugreifen.

Fortsetzung siehe Rückseite

Information Disclosure

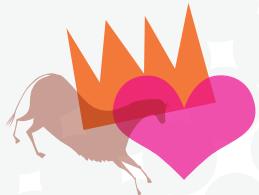


Denial of Service

2. Ein Angreifer kann Ihr Authentisierungs-System unbrauchbar oder unverfügbar machen.
3. Ein Angreifer kann einen Client unverfügbar oder unbrauchbar machen, aber das Problem verschwindet, sobald der Angriff aufhört (Client, authentisiert, temporär).
4. Ein Angreifer kann einen Server unverfügbar oder unbrauchbar machen, aber das Problem verschwindet, sobald der Angriff aufhört (Server, authentisiert, temporär).
5. Ein Angreifer kann einen Client unverfügbar machen, ohne dass eine Authentisierung stattgefunden hat. Das Problem verschwindet nach dem Angriff (Client, anonym, temporär).
6. Ein Angreifer kann einen Server unverfügbar machen, ohne dass eine Authentisierung stattgefunden hat. Das Problem verschwindet nach dem Angriff (Server, anonym, temporär).
7. Ein Angreifer kann einen Client unverfügbar machen, und das Problem besteht fort, nachdem der Angriff aufgehört hat (Client, authentisiert, persistent).

Fortsetzung siehe Rückseite

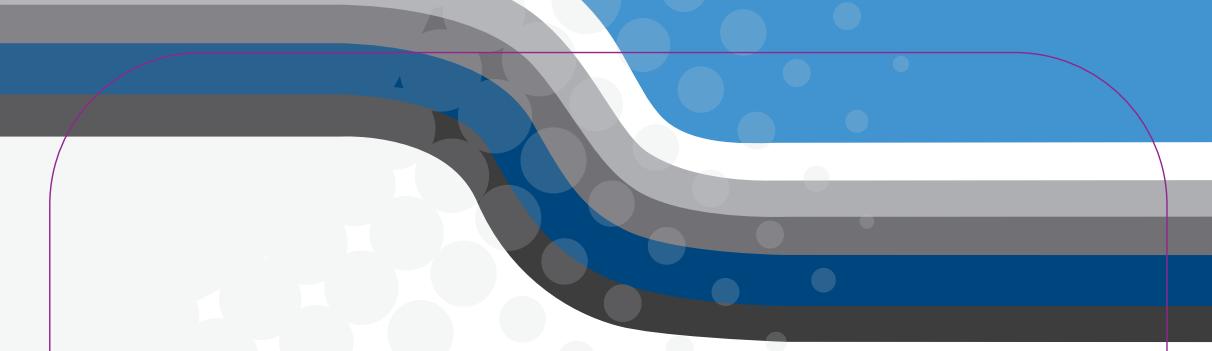
Denial of Service



Elevation of Privilege (EoP)

5. Ein Angreifer kann Einfluss darauf nehmen, welche Art Validierung Daten durchlaufen, die jeweils unterschiedliche Ergebnisse liefern.
6. Ein Angreifer kann Berechtigungen, die Ihr Programm verlangt, aber nicht wirklich benötigt, für sich ausnutzen.
7. Ein Angreifer kann einen Pointer über eine Trust Boundary hinweg angeben, anstatt Daten eingeben zu müssen, die eine Validierung durchlaufen.
8. Eingegebene Daten befinden sich während der Überprüfung noch unter Kontrolle des Angreifers und werden später jenseits der Trust Boundary genutzt.
9. Caller (aufrufende Funktionen) haben keine Möglichkeit, zu überprüfen, welche Validierung Ihr Programm auf die übergebenen Daten anwendet, bevor sie diese weitergeben.
10. Es ist für einen Aufrufer/Caller nicht klar ersichtlich, welche Sicherheitsannahmen Sie treffen.
- J. Ein Angreifer kann Eingaben zum Nutzer zurück spiegeln, z.B. per Cross-Site-Scripting (XSS).
- Q. Sie inkludieren benutzer generierten Inhalt (User Generated Content, UGC) in Ihrer Webseite, der auch Inhalte beliebiger URLs etc. enthalten kann.
- K. Ein Angreifer kann ein Kommando einschleusen, welches vom System mit einer höheren Berechtigung ausgeführt wird.
- A. Sie haben einen neuen Elevation of Privilege Angriff erfunden.

Elevation of Privilege



About

Threat Modeling

Das Elevation of Privilege Spiel ist eine spielerische und schnell zu erlernende Methode, Ihr Projekt-Design aus der Security Perspektive zu betrachten.

Es ist eine Methode, Bedrohungsmodelle zu erstellen und dafür gedacht, von Entwicklergruppen aufgegriffen und genutzt zu werden. Es nutzt das STRIDE Modell als Orientierungshilfe.

STRIDE steht für:

Spoofing: Sich als jemand oder etwas ausgeben.

Tampering: Code oder Daten modifizieren.

Repudiation: Eine Aktion abstreiten.

Information Disclosure: Informationen gegenüber jemand preisgeben, der dafür nicht autorisiert ist.

Denial of Service: Dienste stören oder behindern.

Elevation of Privilege: sich Zugang oder Fähigkeiten verschaffen, für die man nicht autorisiert ist.

Unter www.microsoft.com/security/sdl/eop bietet Microsoft Videos, Score Sheets sowie Tips und Tricks für das Spiel an.

About