Measuring the Security of Web Applications

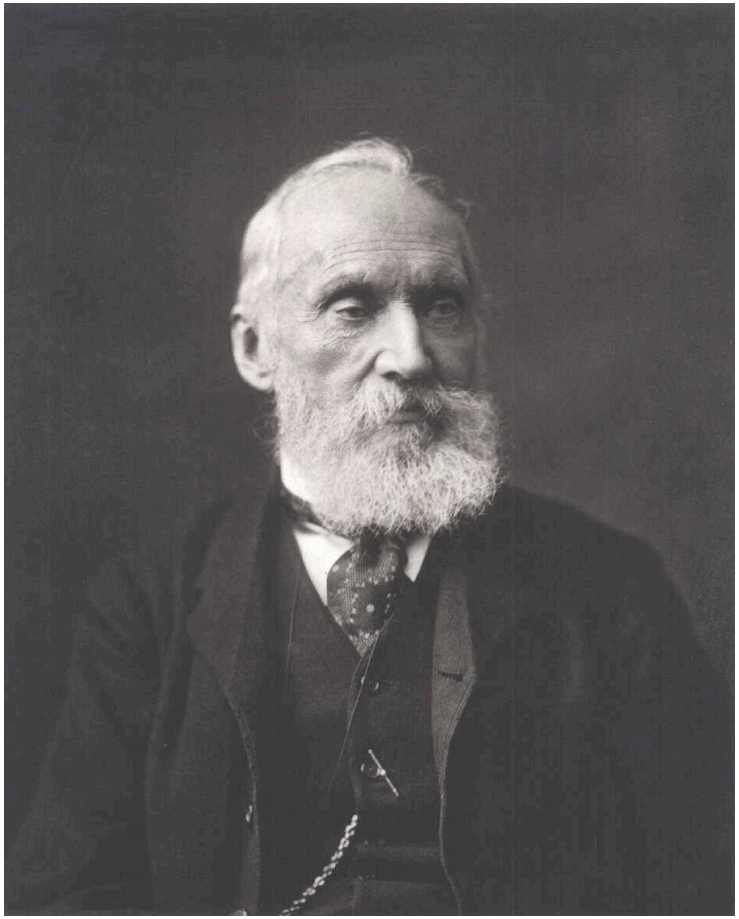
Sebastian Schinzel

Virtual Forge GmbH
Sebastian.Schinzel@virtualforge.de
+49 622 1 868 900

OWASP
Frankfurt, 25.11.08

Daten im Internet

Lord Kelvin 1824 - 1907



"If you can not measure it,
you can not improve it."

"I often say that ***when you can measure***
what you are speaking about,
and express it in numbers,
you know something about it;"

"but ***when you cannot express it in numbers****,*
your knowledge is *of a meagre*
and ***unsatisfactory*** *kind;"*

"it may be the beginning of knowledge,
but ***you have scarcely****, in your thoughts,*
advanced to the stage of science*,*
whatever the matter may be."

http://en.wikiquote.org/wiki/William_Thomson,_1st_Baron_Kelvin

Software ist...

trivial / komplex

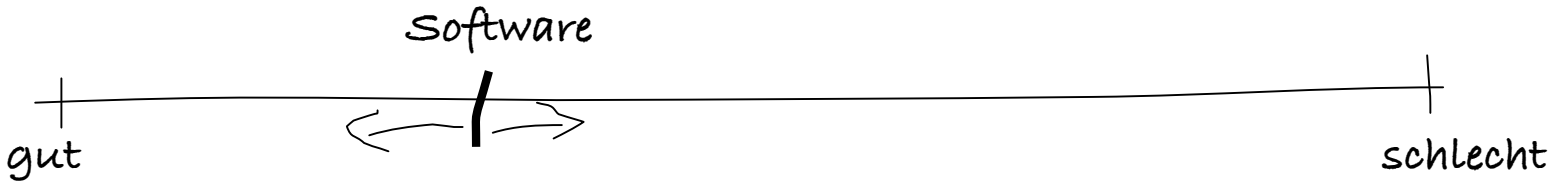
stabil / buggy

günstig / teuer

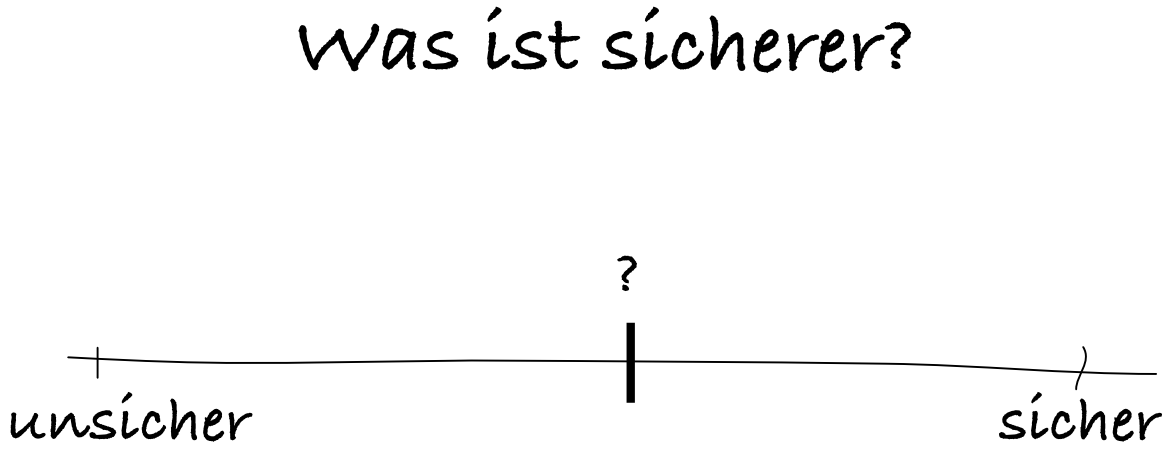
langsam / performant

gut / schlecht

sicher / unsicher



"Mein Apfel ist besser als Dein Fenster"



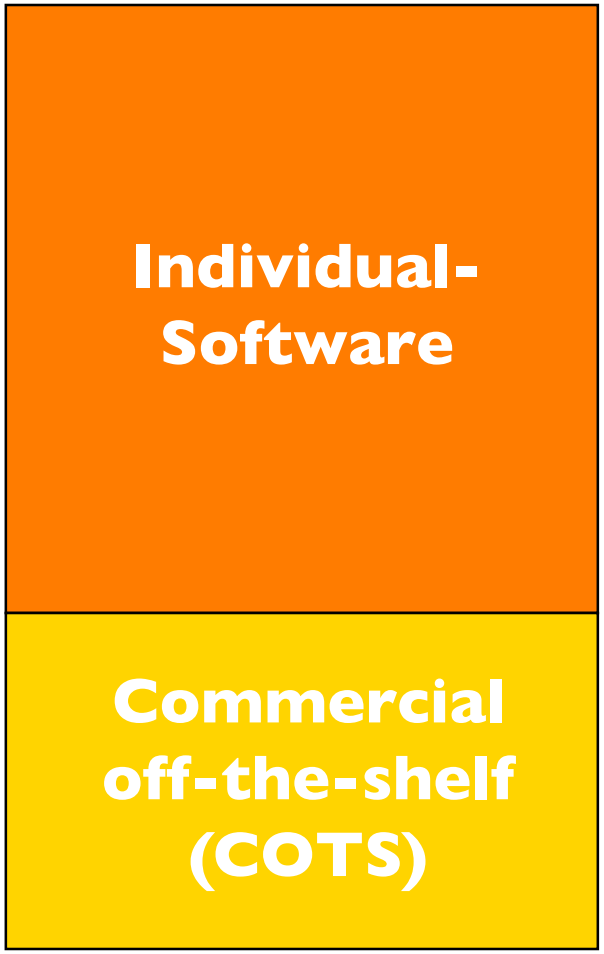
"Von null auf hundert in 25.000€"

Betriebssystem ist "*Unbreakable*"!

"Sicherheit über Nacht mit PowerSecure ™!"

Studie:
"Apfel X ist etwa doppelt so sicher wie Birne Y."

COTS vs. Individualsoftware



Individual-Software:

- Geschäfts-Applikationen, z.B.
  - Web-Shops
  - E-Recruiting
- Web-Applikationen!
- Wartung oft unklar
- Enthält oft kritische und trivial findbare Sicherheits-Lücken

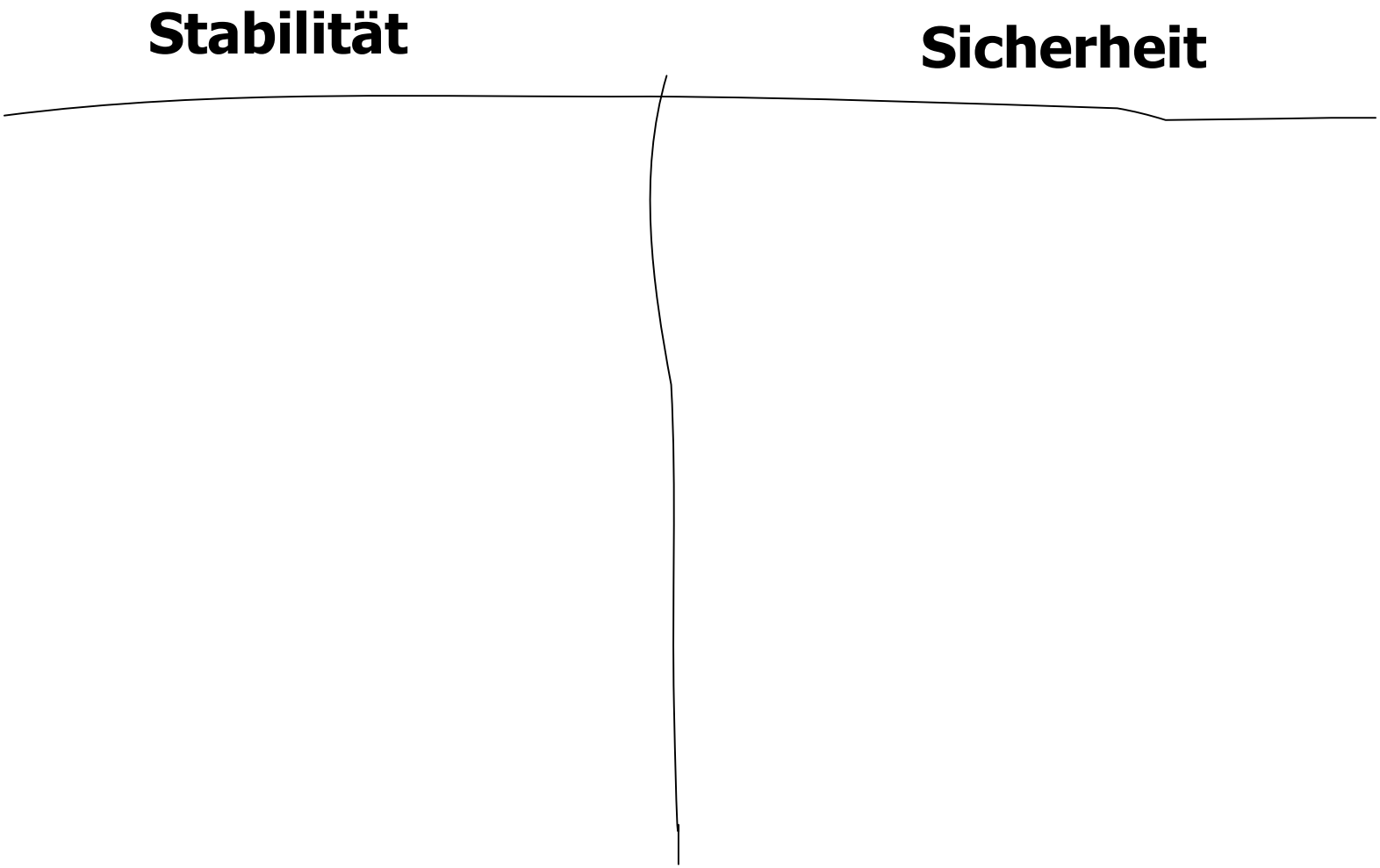
COTS:

- Betriebssystem (Win, Mac, Linux, ...)
- Web-Server (Apache, IIS, ...)
- Wartung durch Hersteller
- Prozesse zur sicheren Entwicklung oft vorhanden

Zwischenstand

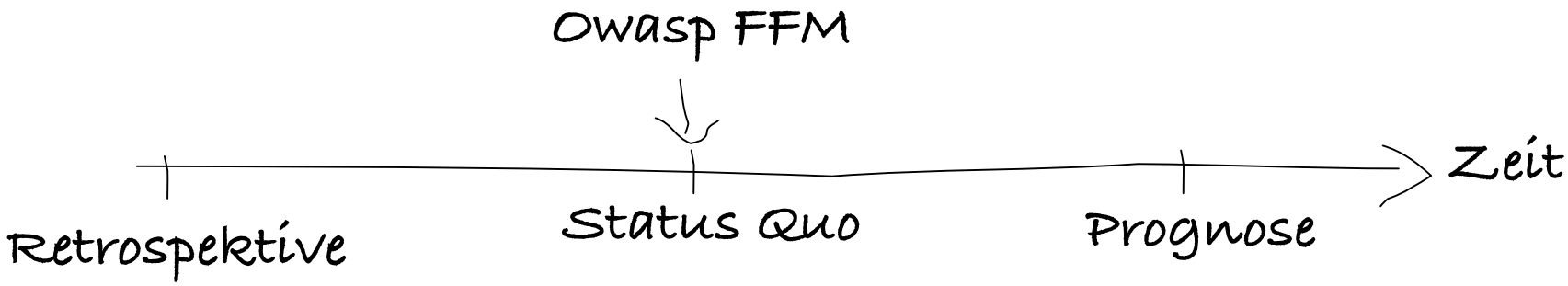
- Daten im Internet müssen geschützt werden
- Sicherheit ist kontra-intuitiv (Marketing hat leichtes Spiel)
- Web-Applikationen haben besondere Anforderungen an Sicherheit

Messen von Software-Eigenschaften



Wie messe ich Sicherheit?

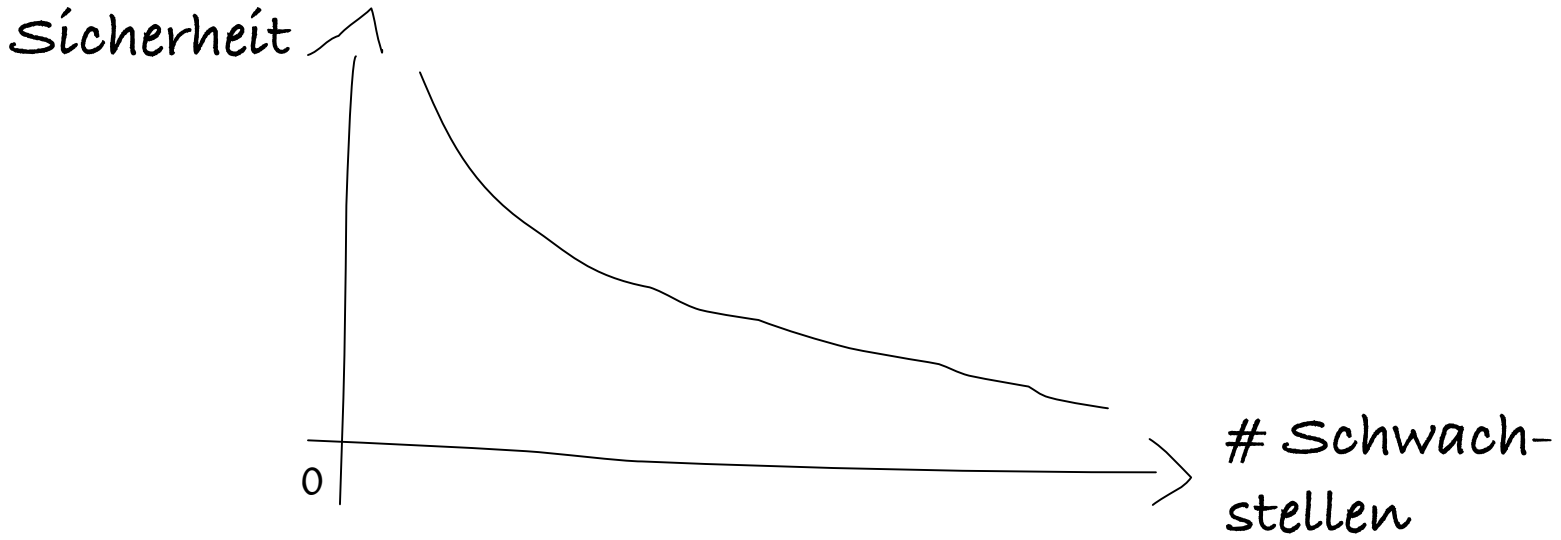
Überlegung: Zählen von Schwachstellen



Wie messe ich Sicherheit retrospektiv?

Daten von Bugtraq, Full-Disclosure, CVE, ...

Annahme: Anzahl veröffentlichter Schwachstellen sei anti-proportional zur Sicherheit



Wie messe ich Sicherheit retrospektiv?

Problem: Veröffentlichung von Sicherheitslücken hat strategischen Wert

- Forscher veröffentlicht Schwachstelle nicht
- Hersteller/Berater veröffentlicht Schwachstelle nicht
- Hacker veröffentlicht Schwachstelle nicht
- Zeitpunkt der Veröffentlichung != Zeitpunkt der Entdeckung

Wie messe ich Sicherheit im Status Quo?

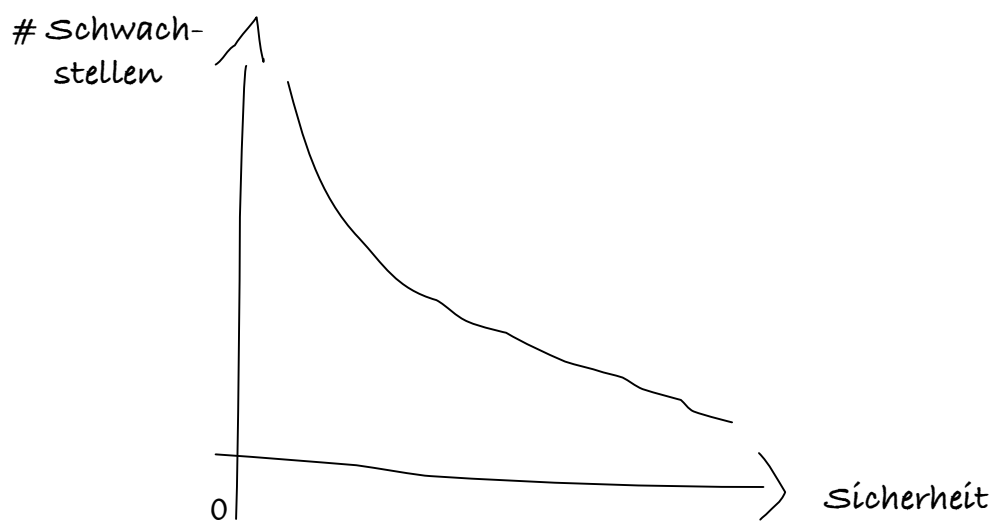
Penetrationstest oder Code Audit:

"Ist ein bestehendes Software-System unsicher?"

- "Ja"

  oder

- "Keine Ahnung"

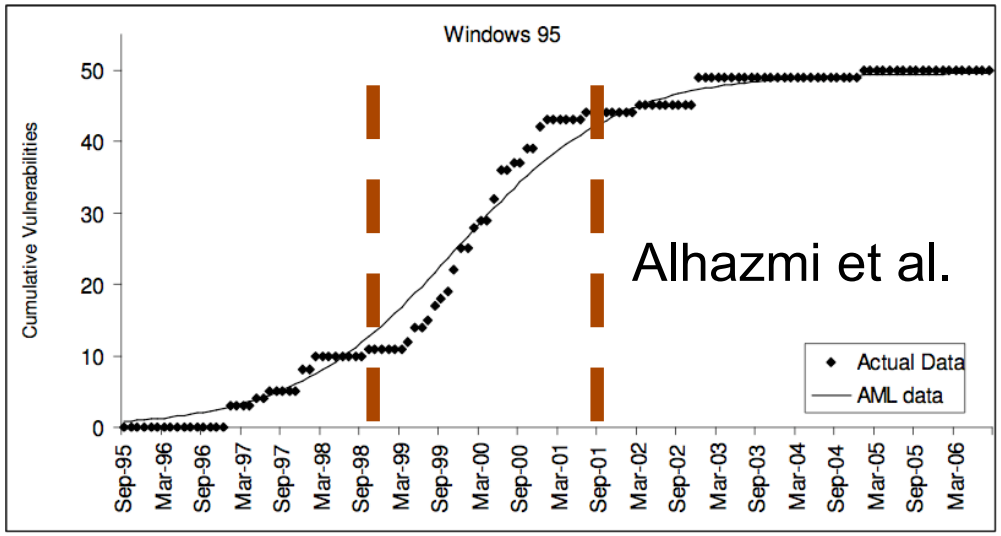


Wie messe ich Sicherheit im Status Quo?

Probleme von Penetrationstests:

- Reproduzierbarkeit?
- Abdeckung?
- Aussage?

Wie messe ich Sicherheit prognostizierend?



Nutzlos wenn:

- # Sicherheitslücken unbekannt
  - Popularität gering
  - Verbreitung gering (Geschäfts-Applikationen?)

Zählen von Schwachstellen...

- Schwachstellen werden nicht immer publiziert
- Individuelle Web-Applikationen haben keine aussagekräftigen Daten über vergangene Schwachstellen
- Messen von Sicherheit schwieriger zu messen als z.B. Stabilität

Zählen von Schwachstellen hat nur wenig Aussagekraft über Sicherheit von Web-Applikationen

Neuer Versuch:

- Suche nach Ursprung typischer Schwachstellen
- Zähle Vorkommen dieser Anti-Patterns

Software-Sicherheit: Wie messen?

Ursprung von Cross-Site-Scripting-Schwachstelle?

Keine ausreichende Trennung zwischen Benutzer-Daten und HTML- oder JavaScript-Kommandos

Eingabevalidierung:

- Kreditkartennummer sollte validiert werden
- "Ist das eine semantisch korrekte Kreditkarten-Nummer?"
- "Darf ein Vorname das Zeichen '<' enthalten?"
- Filtern, Löschen, Ablehnen?

Enkodierung von Datenausgaben:

- Daten müssen als Daten markiert werden
- Ausgaben in HTML:
  - <b> --> <b&gr;
- Ausgaben in JavaScript
  - O'Neil --> O\'Neil
- Ausgaben in URL
  - Ja/Nein --> Ja%2fNein

Software-Sicherheit: Wie messen?

- Sicherheit soll Entscheidungungsgrundlage für Produktwahl sein:

  "Ist Applikation A, oder Applikation B sicherer?"
- Sicherheit verschiedener Systeme kann nur verglichen werden, wenn Sicherheitsanforderungen gleich sind
- Sicherheit kann nur im Vergleich mit Sicherheitsanforderung bewertet werden

Software-Sicherheit: Wie messen?

Beispiel (1): Sicherheitsanforderung: "Trennung von Benutzerdaten und Kommandos bei HTML-Ausgabe"

Konzept: Definition einer Enkodierungs-Funktion f, die Benutzereingaben *A* in HTML-Daten *B* überführt

Validierung:

- Ist *f* korrekt und vollständig? → Audit der Implementierung von *f*
- Wurden alle HTML-Ausgaben bedacht? → "Anteil der HTML-Ausgaben ohne Enkodierung durch *f*"
- Approximierung: Aufruf von f muss innerhalb von 5 Zeilen vor HTML-Ausgabe erfolgen
- Suche kann mit Source-Code-Scanner automatisiert werden

Software-Sicherheit: Wie messen?

Beispiel:

OK:

```
String str = request.getParam("x");
str = f(str);
response.write("<b>" + str + "</b>");
```

Nicht OK:

```
String str = request.getParam("x");
response.write("<b>" + str + "</b>");
```

Software-Sicherheit: Wie messen?

Beispiel:

Nicht OK:

```
String str = request.getParam("x");

   ...

if(immer_wahr) {
   str = f(str);
}

   ...

response.write("<b>" + str + "</b>");
```

Software-Sicherheit: Wie messen?

- Aufwand um Sicherheitsanforderung zu erfüllen?
- Entwickler benötigt 0,1 Stunden, um HTML-Ausgabe mit Aufruf von *f* zu versehen

Applikation A:

- 100 HTML-Ausgaben ohne Aufruf von *f*: $C_{A1}=100*0,1h=10h$
- 1 Fehler in *f*: $C_{A2}=8h$
- $C_A=C_{A1}+C_{A2}=\mathbf{18h}$

Applikation B:

- 1000 HTML-Ausgaben ohne Aufruf von *f*: $C_{B1}=1000*0,1h=100h$
- 0 Fehler in f: $C_{B2}=0h$
- $C_B=C_{B1}+C_{B2}=\mathbf{100h}$

Software-Sicherheit: Wie messen?

"Um Applikation A sicher gemäß der Sicherheitsanforderung zu machen, ist ein Aufwand von 18 Entwicklerstunden nötig."

"Um Applikation B sicher gemäß der Sicherheitsanforderung zu machen, ist ein Aufwand von 100 Entwicklerstunden nötig."

→ Gemäß der Sicherheitsanforderung ist Applikation A zu bevorzugen

Zusammenfassung

Software-Sicherheit im Web ist kritisch

Sicherheit ist kontra-intuitiv. Man muss messen, um wirklich verbessern zu können

Zählen von Schwachstellen hat nur wenig Aussagekraft bei Web-Applikationen

Möglicher Weg: Leicht messbare Software-Qualitäts-Anforderungen mit direktem Einfluß auf Software-Sicherheit

Danke für die Aufmerksamkeit!

Kommentare

Anmerkungen

Kritik

Diskussion

Fragen

Sebastian.Schinzel@virtualforge.de