# Hva skjer i OWASP?

**Kåre Presttun**
**Chapter Lead**
**Mnemonic as**
kaare@mnemonic.no
+47 4100 4908

**OWASP**
Norway Chapter

Copyright 2007 © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.

**The OWASP Foundation**
http://www.owasp.org

# Agenda

- Hva er OWASP
- Prosjekter i OWASP
- Konferanser

# Først – Hva er OWASP?

- OWASP er et åpent fellesskap som har som formål å gjøre det mulig å utvikle, kjøpe og vedlikeholde web-applikasjoner som en kan stole på.
- Alle OWASP verktøy, dokumenter og forum er gratis og alle interesserte kan delta.
- Arbeidet foregår i prosjekter (49 – vi skal ikke se på alle).
- Medlemsavgift går til å finansiere prosjekter.
- Informasjon spres på konferanser
- Informasjon spres på møter i ”Local Chapter” som det finnes 113 stykker av (Sverige, Danmark, Finnland)
- Det var stiftelsesmøte for Norway Chapter onsdag 2. april og generalforsamling mandag 28. april

# OWASP Top Ten Project

- A1 - Cross Site Scripting (XSS)
- A2 - Injection Flaws, particularly SQL injection
- A3 - Malicious File Execution
- A4 - Insecure Direct Object
- A5 - Cross Site Request Forgery (CSRF) (XSRF)
- A6 - Information Leakage and Improper Error Handling
- A7 - Broken Authentication and Session Management
- A8 - Insecure Cryptographic Storage
- A9 - Insecure Communications
- A10 - Failure to Restrict URL Access

Top Ten inneholder detaljert informasjon om hvordan de forskjellige problemene kan fikses og mye referanser

# OWASP AppSec FAQ Project

- FAQ en vinklet mot utviklere.
- Utviklere vil finne korte svar på de vanligste spørsmål.
- For utfyllende forklaringer anbefales OWASP Guide

# OWASP Guide Project

- Et dokument på over 300 sider som er en veiledning i hvordan man utvikler sikre web-applikasjoner og web-services
- Et løpende prosjekt – oppdateres hele tiden
- Dekker ”rike applikasjoner” som Ajax
- Dekker hele utviklingsprosessen som policy, sikker koding, trussel og risikomodellering til konfigurasjon, utrulling og vedlikehold
- Dekker i detalj mange tema som autentisering, autorisering, sesjonshåndtering, datavalidering etc.

# OWASP Testing Guide

- Guiden går inn på hvorfor testing er viktig gjennom hele prosessen fra utvilklingen starter til en er i produksjon og vedlikehold.
- Det finnes detaljert veiledning, på er mengde problemområder så som XSS, om hvordan det kan testes
- Lag testene i parallell med koden
- Veiledning i test av SSL/TLS (normalt feil satt opp)
- Det er også veiledning i testing av web-services og Ajax
- God oversikt over testverktøy, både gratis og kommersielle.

# OWASP WebGoat Project

- Dette er en usikker web-applikasjon som er skrevet for å lære om web-applikasjonssikkerhet.
- Skrevet i Java
- Interaktiv opplæringsmiljø
- Inneholder over 30 øvelser hvor en lærer hvordan de forskjellige typer feil oppfører seg og som en selv kan teste
- Inneholder: Cross Site Scripting, Access Control, Thread Safety, Hidden Form Field Manipulation, Parameter Manipulation, Weak Session Cookies, Blind SQL Injection, Numeric SQL Injection, String SQL Injection, Web Services, Fail Open Authentication, Dangers of HTML Comments, +++

# OWASP WebScarab Project

- Et rammeverk for analyse og testing av applikasjoner som bruker http og https.
- Skrevet i Java
- Den vanligst bruken er som lokal proxy på testeren sin maskin

- Paros og Burp suite er lignende verktøy som også er gratis. Alle har litt forskjellige egenskaper.

# OWASP CAL9000 Project

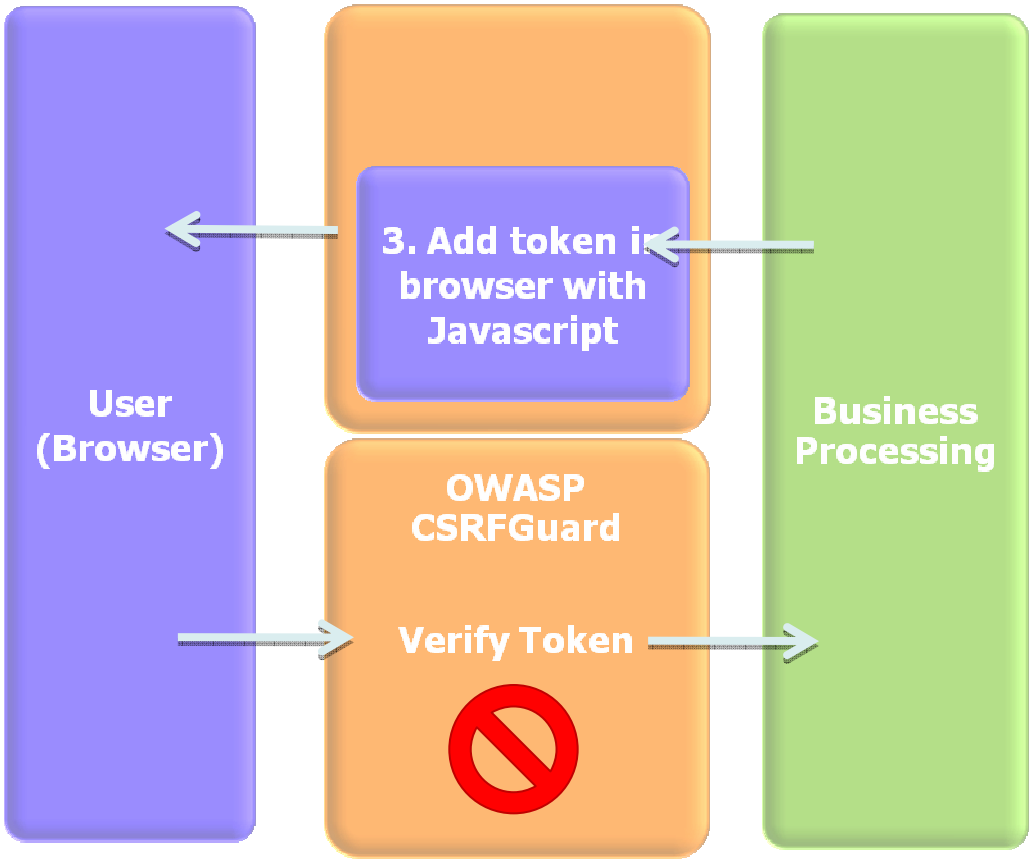
- En samling av av web-applikasjon testverktøy.
- Skrevet i JavaScript (eksekverer i browseren)
- Benyttes gjerne sammen med andre verktøy som OWASP WebScarab og sårbarhetsskannere

# OWASP Stinger Project

- En slags applikasjonslagsbrannmur eller input-filter som legges foran applikasjonen.
- Enkelt å integrere med eksisterende applikasjoner
- Skrevet i Java
- Konfigureres med en XML-fil som beskriver filterreglene

- V 3: Forenkling av regelgenerering med ”learning mode”

# OWASP CSRFGuard Project




- Token må være kryptografisk sikkert
- adderer token til:
  - href attributt
  - src attributt
  - hidden field i alle forms
- Aksjoner
  - logge
  - redirigere

- OWASP CSRF Tester

# OWASP AntiSamy Project

- Å la brukerne legge inn html-kode i en applikasjon er potensielt farlig. Mange husker gjestebøker fra rundt når 2000 som tillot det. Det poppet opp skript som sendte folk rundt til alt mulig, stort sett porno.
- MySpace tillater innlegging av ganske mye men sperrer en del. Det var noen feil i implementeringen.
- Samy ville ha mange venner på MySpace. Han klarte å lage et skript som kom seg gjennom filteret og etter mindre enn 20 t hadde han over 1 000 000 venner. En time etterpå ble MySpace tatt ned for opprydning.

- Les den sprø historien fra Samy selv: http://namb.la/popular/

# OWASP AntiSamy Project

- Hendelsen førte til at en begynte å undersøke muligheten for å lage en generisk filterkomponent som kan beskytte mot dette.
- AntiSamy var født.
- Finnes i Java. PHP og .NET kommer i løpet av 2008. Konfigureres med en XML-fil
- Fire XML-filer følger med (med forskjellig nivå av paranoia):
- antisamy-slashdot.xml
- antisamy-ebay.xml
- antisamy-myspace.xml
- antisamy-anythinggoes.xml

- Kan ta utgangspunkt i en av de og redigere til ønsket policy
- MySpace bruker ”black listing”, AntiSamy bruker ”white listing”

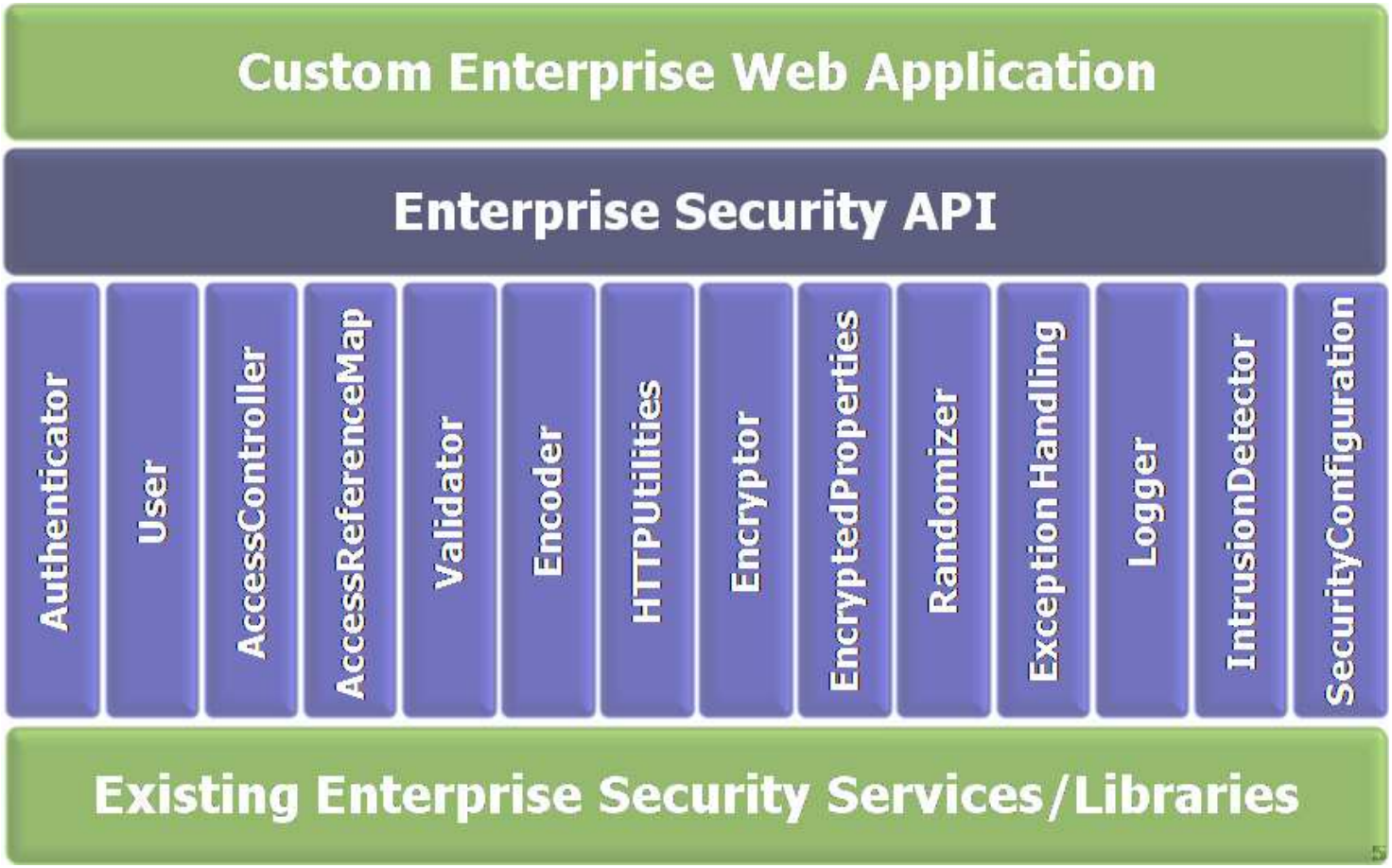
# Enterprise Security API (ESAPI) Project

- En åpen samling av alle sikkerhetsmetodene en utvikler har behov for
- Eksisterende rammeverk og plattformer (Java EE, Struts, Spring, etc...) har noe støtte for sikkerhet men ikke nok
- En åpen referanseimplementasjon
- Antisamy er nå integrert i ESAPI
- Skrevet i Java men .NET og PHP er under arbeid

# Enterprise Security API (ESAPI) Project

- Hvorfor ESAPI
  - Det er vanskelig å lage alle sikkerhetsfunksjonene riktig (krever inngående forståelse av angrepene)
  - Det er ofte ikke nok tid i prosjektene til å gjøre det riktig
  - Utviklerne skal ikke bruke tiden på å lage alle sikkerhetsfunksjonene (stor forskjell på å lage og å bruke)
  - Utviklerne skal bare bruke sikkerhetsfunksjonene
  - Lage en gang, bruke igjen og igjen (de fleste trenger det samme)

# Enterprise Security API (ESAPI) Project



# Enterprise Security API (ESAPI) Project

| OWASP Top Ten | OWASP ESAPI |
|---|---|
| Cross Site Scripting | Validator, encoder |
| Injection Flaws | Encoder |
| Malicious File Execution | HTTPUtilities (upload) |
| Insecure Direct Object | AccessReferenceMap |
| Cross Site Request Forgery | User (csrftoken) |
| Information Leakage, Error Handling | EnterpriseSecExeption HTTPUtils |
| Authentication, Session Management | Autheticator, User, HTTPUtils |
| Insecure Cryptographic Storage | Encryptor |
| Insecure Communications | HTTPUtils (sec. cookie, channel) |
| Failure to Restrict URL Access | AccessController |

# OWASP AppSec konferanser

- Nettopp ferdig, Europa
  - OWASP AppSec Europe 2008 - Ghent, Belgium
  - May 19th - 22nd - Conference & Training
- Kommer, US
  - OWASP AppSec U.S. 2008 - New York City
  - September 22nd - 25th - Conference & Training
- Kommer
  - OWASP AppSec India 2008 - Delhi, India, August 20th - 21st
  - OWASP Israel 2008 - Herzliya, Israel, September 14th
  - OWASP AppSec Europe 2009 - Krakow, Poland

# Oppsummering

- OWASP har en mengde verktøy og veiledninger som kan hjelpe deg til å utvikle sikre web-applikasjoner
- Det er bedre å integrere inn en komponent fra OWASP enn å finne opp hjulet på nytt
- Og hvorfor ikke bidra til å forbedre en av de eksisterende modulene så den passer bedre i ditt prosjekt?

*Takk for oppmerksomheten*