Secure Coding Dojo

Making Software Security Training Fun

TM

About me

- Security Architect and R&D Security Leader at Trend Micro
- Passionate about software security
- OWASP Ottawa Chapter Co-Lead
- OWASP Secure Coding Dojo Project Lead

Paul Ionescu

@pentesq

Why Software Security Training Matters?

- **Awareness** - Get development on your side
- **Prevention** - Works before the code is written
- **Coverage** – Empower development teams to conduct security activities
- **Visibility** – Can't know everything that is being built, but your development advocates will let you know when they need your help
- Software Security Training is the **first and most important step** in the AppSec journey

Quiz Time!

The 3 ways of DevOps are: Systems Thinking, Amplify Feedback Loops and _____

a) Implement A/B Testing
b) Site Reliability Engineering
c) Integrated Project Management
d) Culture of Continual Experimentation and Learning

Software Security Learning is fundamental to the 3rd Way of DevSecOps

Challenges Teaching Software Security

- **Presentations can be boring** and the information does not persist after the training is over

- What if someone was sick, missed the training…**will they really watch the recording**? How about new hires? How about remote teams?
  - Do you ever find yourself drifting into your e-mail while watching a recording?

- Engaging presenters can be very effective but **not all security people are engaging presenters**

- **Difficult to collect metrics** and understand organization coverage from a presentation

Robin Higgins / pixabay

Gamification of Training

- Developers love **puzzles and games**
- Games **stimulate the mind**, removing boredom
- **Competitions** drive increased adoption
- **Learn by doing**. If the participants conduct a Cross-Site Scripting attack they understand it better.

Robin Higgins / pixabay

Can you find the Cross-Site Scripting?

Q4: The following example is showing the use of a JavaScript parameterized statement. Can you spot the snippet that leverages this best practice?

```
$get("/profile", function(data, status){
    if(data!=null){
        var dataArgs = data.split(",");
        if(dataArgs.length > 1){
            var displayName = dataArgs[0];
            setTimeout(`showProfile('${displayName}')`, 1000);
        }
    }
});
```

```
$get("/profile", function(data, status){
    if(data!=null){
        var dataArgs = data.split(",");
        if(dataArgs.length > 1){
            var displayName = dataArgs[0];
            setTimeout(showProfile, 1000, displayName);
        }
    }
});
```

Taken from one of the Dojo challenges

Capture the Flag

- **Capture the Flag (CTF)** - events where security professionals prove their skills by hacking vulnerable systems

- OWASP **vulnerable software/CTF projects** –Dev Slop, Juice Shop, Security Shepherd, Web Goat

- CTFs are great for AppSec professionals and pen-testers, **not a good fit for development**
  - Developers that enjoy CTFs, **< 30%** of the organization
  - In some CTFs **not everyone gets the same experience**, participants join teams and each picks a different challenge
  - **CTFs focus on challenge difficulty**, not on teaching concepts. You may have a CTF with 10 challenges but three different XSS levels, four different authentication levels, and three more injection levels. Does not cover all software security flaws

Requirements

- Provide detailed information about software **weaknesses**, **attacks** and **defenses**.
- Practice the attacks just like in a CTF challenge, but the **attacks would be easy to conduct**
  - No special tools needed, **only a browser**
- Comprehensive training curriculum based on **SANS Top 25** and **OWASP Top 10**
- **Self-paced**, complete one lesson at a time, based on availability
- **Always available**, new developers joining the team could take the training as an on-boarding activity
- Leaderboard, the **training would be a game**. Multiple levels, awards, badges.

Welcome to Secure Coding School!

- Inspired from Karate
- Started in 2017
- Open Source, Apache 2 License
- 3 Different Learning Modules
  - Black Belt (2017)
  - Second Degree Black Belt (2018)
  - Security Code Review Master (2019)
- 34 lessons

Training Material

- **Black Belt**
  - This module is based on the SANS Top 25 - Most Dangerous Software Flaws. Lessons are entry level difficulty aimed at introducing the concepts of vulnerability, exploit and software defense.

- **Second Degree Black Belt**
  - CTF like module, based on OWASP Top 10 (v2017). Participants take down the cloud applications used in a worldwide malware campaign.

- **Security Code Review Master**
  - Developers learn to apply security elements to code review.

Training Syllabus – Black Belt (Insecure.Inc)

Challenge Name	SANS 25 CWE(s)	OWASP Top 10 2017	PCI-DSS Req. 6
Yellow Belt : Missing Authentication for Critical Function	CWE 306	A2	6.5.10, 6.5.8
Yellow Belt : Reliance on Untrusted Inputs in a Security Decision	CWE 807	A2; A5	6.5.10. 6.5.8
Yellow Belt : Missing Authorization	CWE 862	A5	6.5.10
Orange Belt : Missing Encryption of Sensitive Data	CWE 311	A3	6.5.3, 6.5.4
Orange Belt : Use of a Broken or Risky Cryptographic Algorithm	CWE 327	A3	6.5.3, 6.5.4
Orange Belt : Use of a One-Way Hash without a Salt	CWE 759	A3	6.5.3, 6.5.4
Green Belt : Password Guessing Attack	CWE 307; CWE 798	A2	6.5.10
Green Belt : Integer Overflow or Wraparound	CWE 190	N/A	N/A
Green Belt : Download of Code Without Integrity Check	CWE 494	N/A	N/A
Purple Belt : URL Redirection to Untrusted Site ('Open Redirect')	CWE 601	N/A	N/A
Purple Belt : Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') and related flaws	CWE 79; CWE 829	A7	6.5.7
Purple Belt : Cross-Site Request Forgery (CSRF)	CWE 352	N/A	6.5.9

Challenge Name	SANS 25 CWE(s)	OWASP Top 10 2017	PCI-DSS Req. 6
Blue Belt : Unrestricted Upload of File with Dangerous Type	CWE 434	N/A	6.5.8
Blue Belt : Improper Restriction of XML External Entity Reference ('XXE')	CWE 611	A4	6.5.1
Blue Belt : Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	CWE 22	A5	6.5.8
Brown Belt : Incorrect Authorization	CWE 863	A5	6.5.4
Brown Belt : Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') and related flaws	CWE 78; CWE 250; CWE 732	A1	6.5.1
Brown Belt : Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	CWE 89	A1	6.5.1, 6.5.5
Black Belt : Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') and related flaws	CWE 120; CWE 676	N/A	6.5.2
Black Belt : Use of Externally-Controlled Format String	CWE 134	N/A	N/A
Black Belt : Quiz	All of the above	All of the above	All of the above

Demo - Lesson Experience

Black Belt – Buffer Overflow Challenge

2nd Degree Black Belt (Hacker's Den)

Challenge Name	SANS 25 CWE(s)	OWASP Top 10 2017	PCI-DSS Req. 6
Security Misconfiguration	N/A	A6	N/A
Sensitive Data Exposure	CWE 311; CWE 327; CWE 759	A3	6.5.3, 6.5.4
Broken Authentication & Broken Access Control	CWE 306; CWE 862	A2; A5	6.5.10, 6.5.8
Cross-Site Scripting	CWE 79	A7	6.5.7
Injection	CWE 78	A1	6.5.1
XML External Entities	CWE 611	A4	6.5.1
Using Components with Known Vulnerabilities & Insecure Deserialization	CWE 509	A8; A9	6.5.1

Training Syllabus – Security Code Review Master

Challenge Name	SANS 25 CWE(s)	OWASP Top 10 2017	PCI-DSS Req. 6
Input Validation	Various	Various	Various
Parameterized Statements	CWE 78; CWE 89;	A1	6.5.1
Memory Best Practices	CWE 120; CWE 131; CWE 193; CWE 134	N/A	6.5.2
Protecting Data	CWE 311; CWE 312; CWE 759; CWE 319; CWE 327	A3	6.5.3, 6.5.4
Preventing Cross-Site Scripting	CWE 79;	A7	6.5.7
Indirect Object References	CWE 22; CWE 601	A5	6.5.8

Demo - Lesson Experience

Security Code Review Master – Memory Best Practices

Managing Security Training for a Large Org.

- Authentication with LDAP/SAML
  - The Dojo integrates with ADFS SAML, LDAP, Slack Auth, Google Auth
- Dashboards and metrics
  - Team stats and overall organization stats
- Reports
  - Generate completion status reports based on a CSV

Module Name	Player Count
Black Belt	2176
Second Degree Black Belt	726
Security Code Review Master	249

Demo – Teams and Reports

Running the Secure Coding Dojo

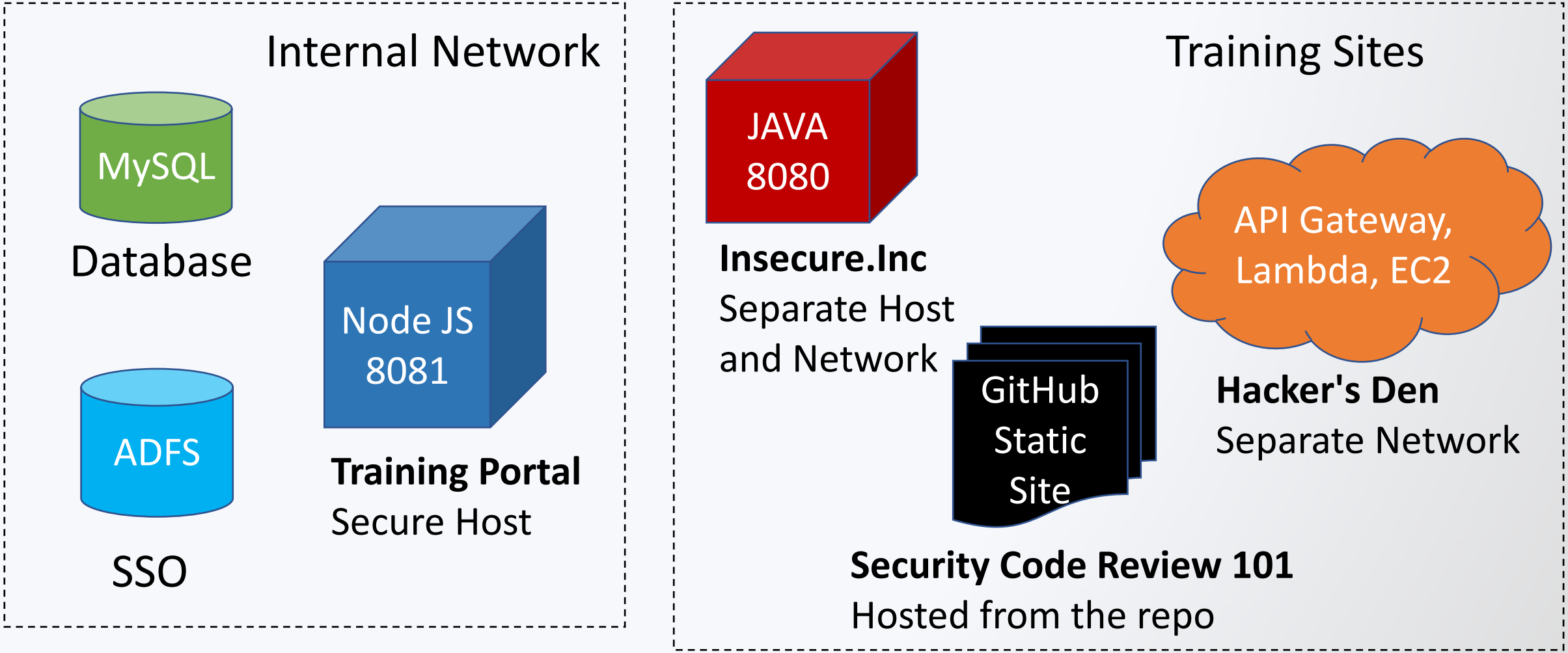
- Really easy with docker. Docker images are published under: https://hub.docker.com/u/securecodingdojo

- Quick setup:

```
git clone https://github.com/trendmicro/SecureCodingDojo.git
export DATA_DIR = ~/dojofiles
docker-compose up
```

- Production config, building your own VM and more on the project wiki: https://github.com/trendmicro/SecureCodingDojo/wiki

Secure Coding Dojo Deployment



Future Plans

- More participation
- Translations
- Investigate Integrations with other OWASP vulnerable applications: Juice Shop, Dev Slop
- Containerization of Second Degree Black Belt
- New modules
- Roles
- Reporting

Contribute

- Use the Dojo!
- Pull requests are welcome!
  https://github.com/trendmicro/SecureCodingDojo
- Twitter: @SecureCodeDojo
- OWASP Global Slack: #secure-coding-dojo

Q&A

Rate this Session

SCAN THE QR CODE TO COMPLETE THE SURVEY

Secure Coding Dojo

Making Software Security Training Fun

Thank You!