Automated Web Foo or FUD?

David Kierznowski
IT Security Analyst
david.kierznowski@gmail.com

OWASP Day

Belgium

6 Sep 2007

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

Powered by...

http://www.nta-monitor.com

About DK

- Check Team Leader with UK's CESG Scheme
- Senior Security Analyst for a leading penetration testing company in the UK
- Works in both government and commercial sectors
- Core member of GNUCITIZEN group
- Developed a number of open source security tools (i.e. ASP-Auditor, TSF)
- Founder of BlogSecurity, michaeldaw.org and primary developer of wp-scanner
- Credited on several major web application vulnerability findings
- Research featured on Slashdot, eWeek, SecurityFocus and others

Review

- Current Threats and the Attack Renaissance.
- Insight into the effectiveness of automated tools.
- Test less vs test all debate.
- Hybrid testing with the Technika Security Framework.

Goals

1. Basic understanding of how automated security testing tools are constructed.

2. A greater awareness of the challenges that lie ahead with automated web application tools.

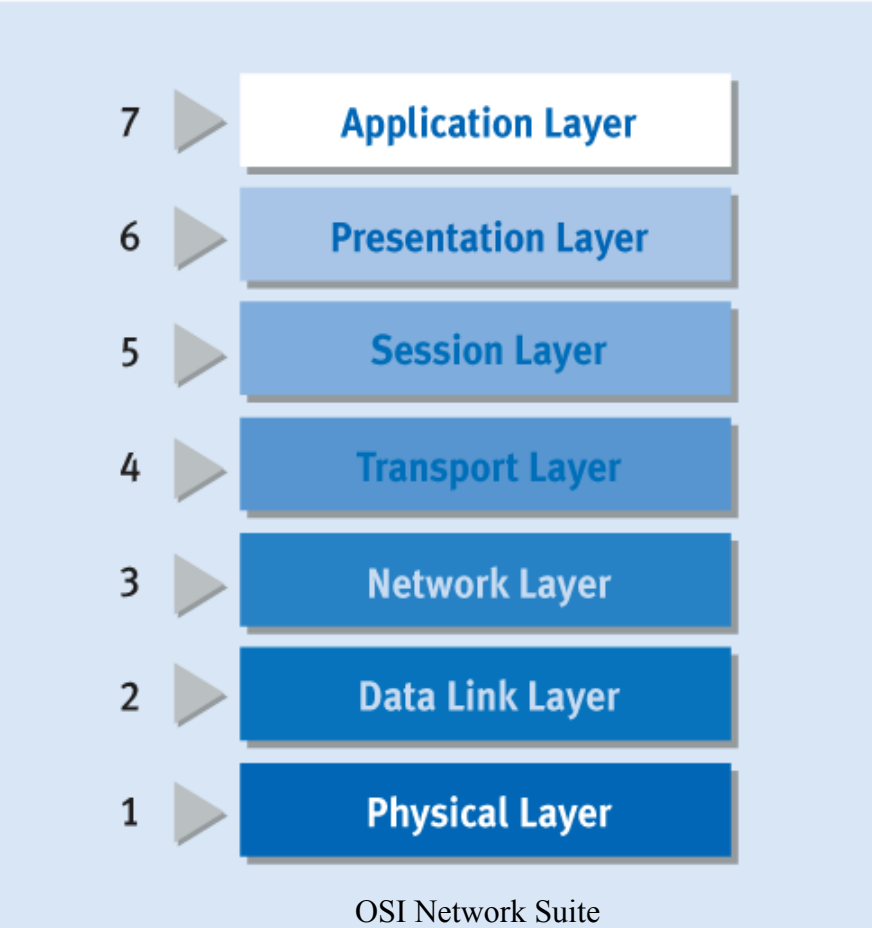
3. An introduction into browser scripting and the Technika Security Framework.

4. A better understanding of how security testing companies approach web application testing.

5. Some suggestions to improve the quality of web application tests.

Current Threats

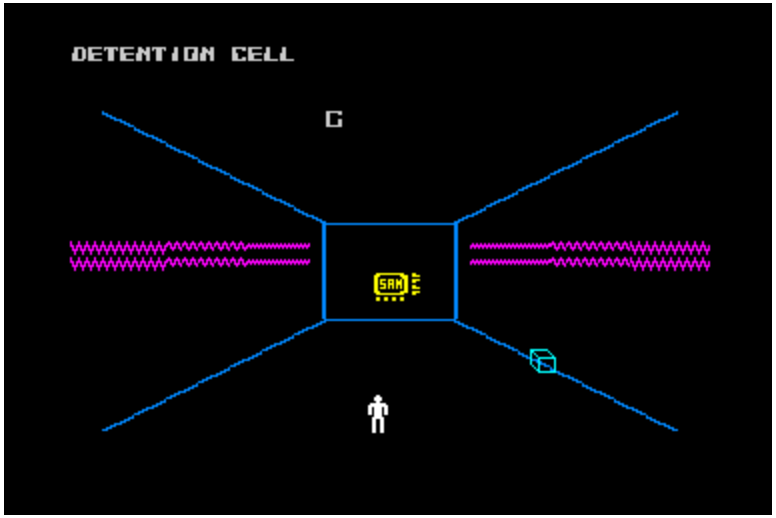
- Gartner Group says 90% of web applications have serious vulnerabilities.
- Symantec says 78% of attacks are at the web application level.
- Mitre last year stated that application-level attacks, such as XSS and SQLI, are replacing buffer overflows as the favourite hacker initiative.



OSI Network Suite

Attack Renaissance

- Traditional vulnerabilities target the server
- New age attacks target the client - XSS as the catalyst



Find Less vs Find All

Find all approach

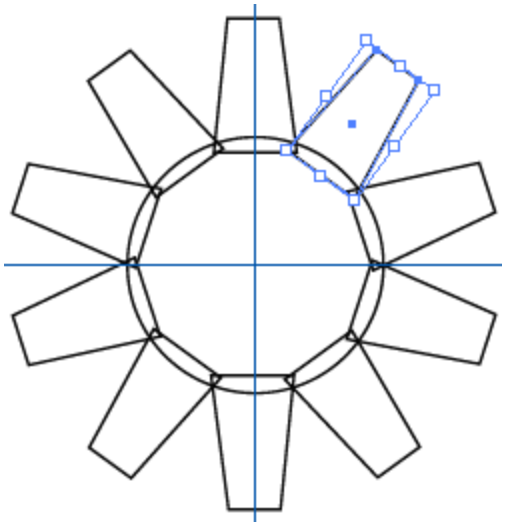
- Try locate and report all vulnerabilities

Find less approach

- Locate less and report the cause of the problem

Simple Automation

Target & Virtual Hosts > Spider & Locate > Sort &Unique > Identify > Test > Result > Report



Auto Limitations “find all”

- Script Parsing
- Business and/or application logic
- False Negatives/Positives
- Non-RFC applications
- State or flow
- Challenge-Response
- Cost
- Custom URLs
- Training
- Denial of Service risks

Manual Limitations "Find less approach"

- Time
- Emotional stability
- Skill level
- Find less approach

The Auto-Worth Model

(F + BA + T) * R = AW

Flow (state)

Business/Application Logic

Technology (i.e. Flash)

Role of web application

A Couple Suggestions

- Define clearly the purpose and role of the application to be tested
- Use the Auto-Worth Model as a base
- Manual security testers should follow a testing procedure
- A hybrid approach may be best

Technika

Automated Browser Exploitation Tool
and
Browser Scripting Tool

Technika – Write Script

- *Snippet from tech.store for persistent storage*

```
tech.store.pop = function(_sarray, key) {

        var _l = _sarray.length;

        for (i=0; i<_l; i++) {
            var _key = key + i;
            sessionStorage.removeItem(_key, _sarray[i]);
        }
        console.log(_sarray.length + " records removed.");

};
```

Technika – Save as bookmarklet

Technika Security Framework

tech.dspider - DOM link spider; because we utilize the DOM, the results are instant.
tech.forms - GET/POST form parser.
tech.mutate - By specifying a payload and regex, we can mutate our target arrays and build tests.
tech.scan - tech.scan is our actual engine that will handle our GET and POST requests.
tech.mNikto - Mini-Nikto was named after the popular web application tool Nikto if you haven't already guessed. We called it mini-nikto as it currently only contains a very small database.
tech.explorer - This is one of my favorite tools in the TS framework. It uses Yahoo! AJAX API (JSON) to fetch links and perform other Google hacking type queries outside of our current DOM. This is really useful even when it is not security related.
tech.store - Utilizes the Firefox sessionStorage to allow us to persistently store arrays.

And much MORE!!

Technika Security Framework

Technika demo video placeholder and tool introduction.

Summary

1. The application layer is enemy number 1!

2. Increased attacks in the future due to the Attack Renaissance

3. Remember the Auto-Worth model in planning and even during the web application testing process.

4. The hybrid approach will almost always present the best results.

5. We hope to start developing Technika to a semi-stable state, but its definitely a project to start getting involved in.

Credits and Refs

Credits:

GNUCITIZEN GROUP – http://gnucitizen.org

NTA-Monitor – http://www.nta-monitor.com

Refs:

Robert Auger - http://www.cgisecurity.com/articles/scannerchallenges.shtml

Technika - http://www.gnucitizen.org/projects/technika/

Technika Security Framework - http://www.gnucitizen.org/blog/introducing-technika-security-framework/

Jeremiah Grossman – Challenges of Web Application Testing