Security Testing : Are We There Yet?

Sajjan Shiva
Department of Computer Science
University of Memphis
Memphis, TN, USA
sshiva@memphis.edu

Outline

1. Current Attack Profile

2. Testing approaches to date

3. Our holistic approach for system building

4. Game Inspired Defense Architecture (GIDA)
    - Our holistic approach for testing

Current Attack Profile

- A considerable amount of work is conducted via the web and more than 80% of attacks occur via the web.
- Infrastructure protection is still needed, but the protection schemes now should concentrate on Applications.
  - 7 out of 10 sites contains SQL injection vulnerabilities.
  - 5 out of 10 sites contain XSS (Cross-site Scripting) vulnerabilities.
- Threats are emergent vulnerabilities – constant monitoring and protection is a must.
- Current software security testing practices are not sufficient.

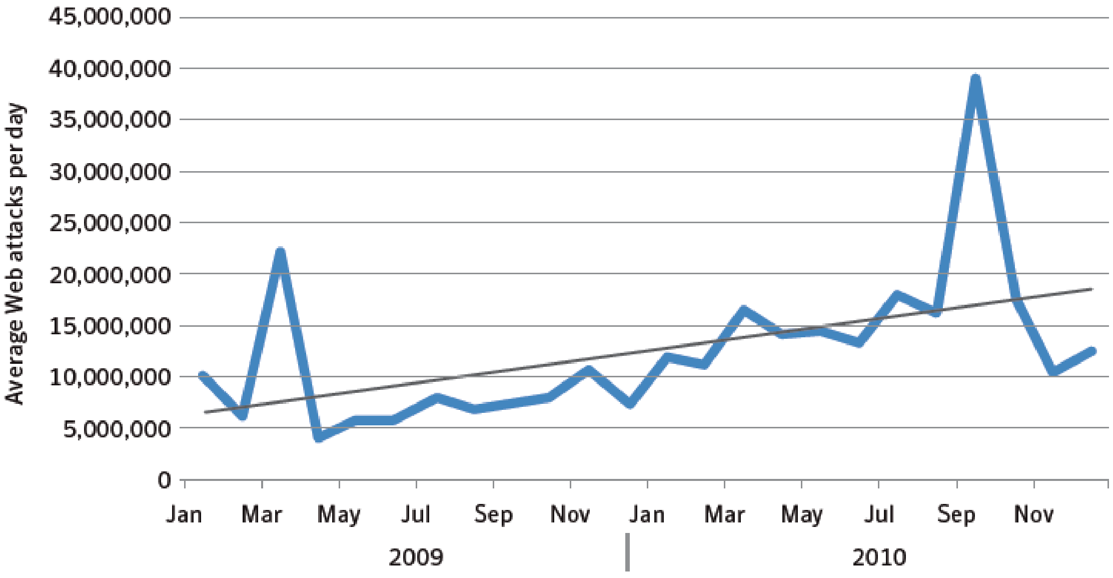
[http://www.net-security.org/secworld.php?id=9880]

[http://www.cisco.com/security]

Current Attack Profile

According to Symantec's Internet Security Threat Report, malware targeting Web browsers and other online applications remains the biggest hazard to enterprise security.

Proliferation of Web attack toolkits drove a 93% increase in the volume of Web-based attacks in 2010 over the volume observed in 2009.

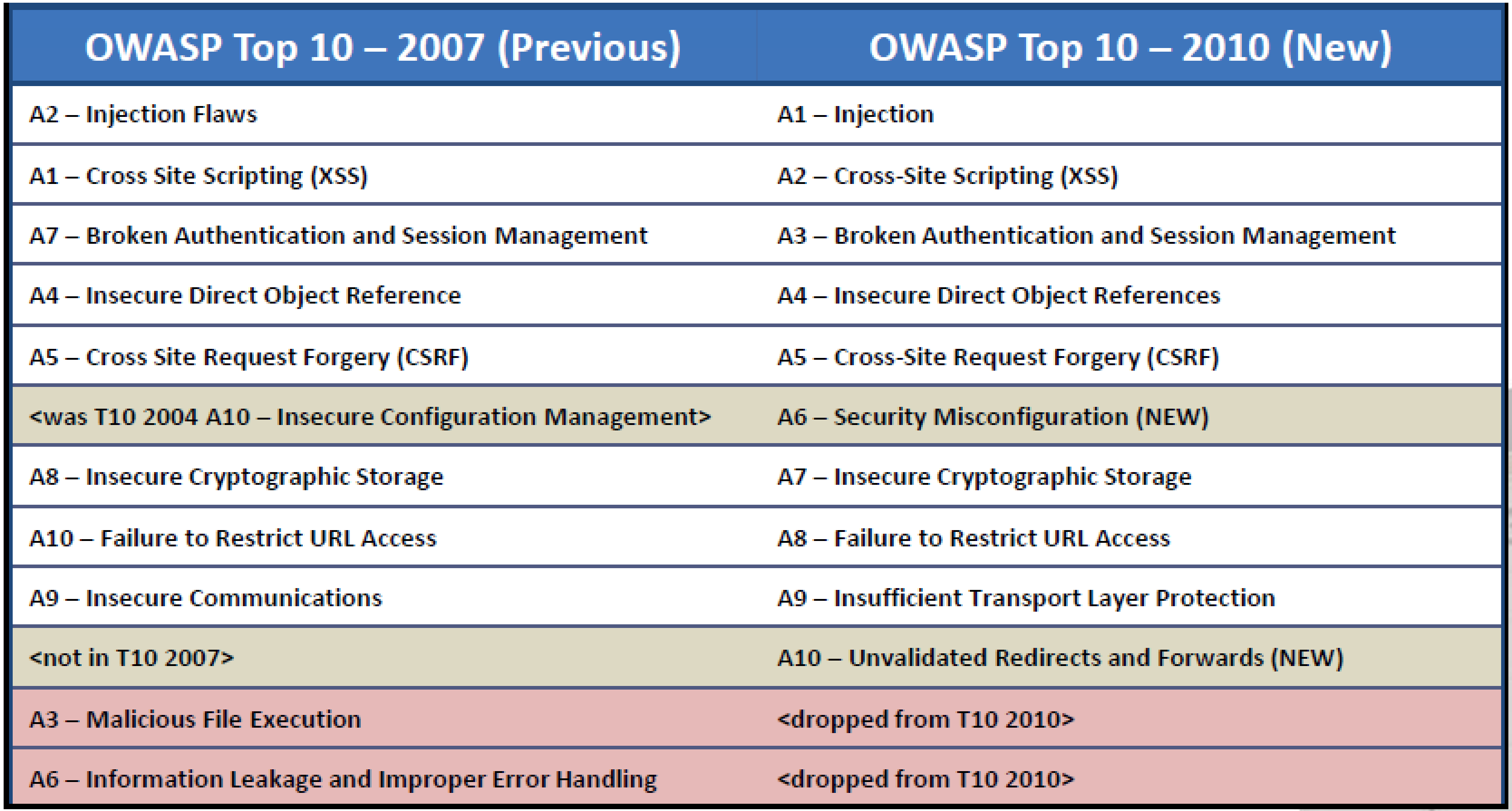


Average Web-based attacks per day, by month, 2009–2010

Source: Symantec Corporation

Source: Symantec's Internet Security Threat Report (2006, 2010)

Top 10 Web Application Security Risks



OWASP Top 10 – 2007 (Previous)	OWASP Top 10 – 2010 (New)
A2 – Injection Flaws	A1 – Injection
A1 – Cross Site Scripting (XSS)	A2 – Cross-Site Scripting (XSS)
A7 – Broken Authentication and Session Management	A3 – Broken Authentication and Session Management
A4 – Insecure Direct Object Reference	A4 – Insecure Direct Object References
A5 – Cross Site Request Forgery (CSRF)	A5 – Cross-Site Request Forgery (CSRF)
<was T10 2004 A10 – Insecure Configuration Management>	A6 – Security Misconfiguration (NEW)
A8 – Insecure Cryptographic Storage	A7 – Insecure Cryptographic Storage
A10 – Failure to Restrict URL Access	A8 – Failure to Restrict URL Access
A9 – Insecure Communications	A9 – Insufficient Transport Layer Protection
<not in T10 2007>	A10 – Unvalidated Redirects and Forwards (NEW)
A3 – Malicious File Execution	<dropped from T10 2010>
A6 – Information Leakage and Improper Error Handling	<dropped from T10 2010>

[https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project]

Trustwave Semiannual Report : The Web Hacking Incident Database

What attack methods do attackers use?

Period: July to December 2010

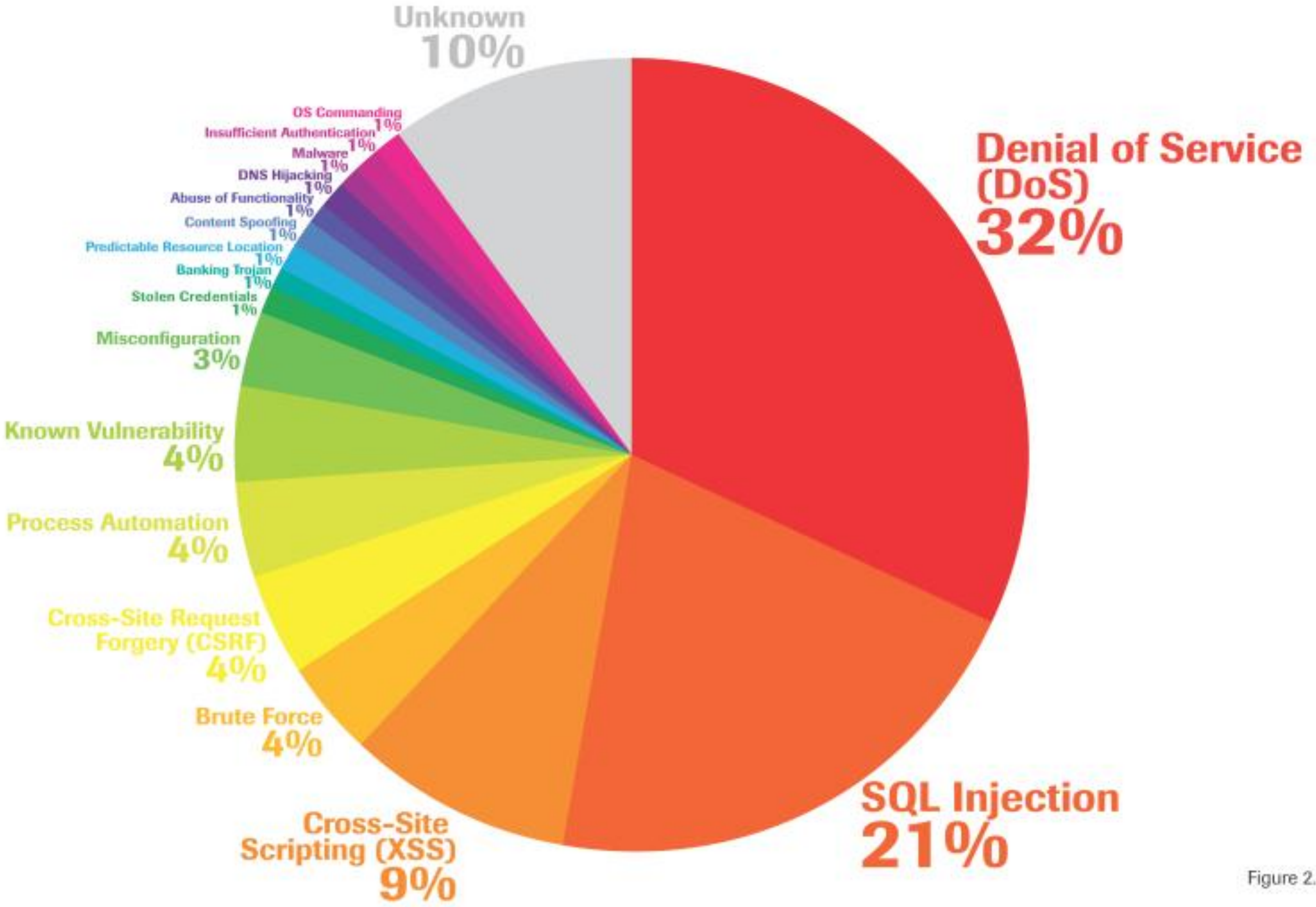


Figure 2.

Testing methodologies To Date:

- Static Code Analysis
  - Performed without actually executing programs.
    - C++: cppcheck, cpplint, PC-Lint, QA-C, etc.
    - Java: Jtest, SonarJ, LDRA Testbed, etc.
- Dynamic Code Analysis
  - Testing and evaluation of a program by executing data.
    - Intel Thread Checker, Intel Parallel Inspector, Parasoft Jtest, VB Watch, etc.
- Penetration Testing
  - Focuses on previously identified risks where probing is conducted to determine if the system is vulnerable.
    - Metasploit Project; w3af (Web Application Attack and Audit Framework)

Limitation of the current secure software testing practices:

- Current testing approaches are largely heuristic, increasingly cumbersome, and are struggling to keep pace with rapidly evolving threats.
- Attacks are more focused towards application level vulnerabilities rather than infrastructure vulnerabilities. Applications evolve rapidly; complexity keeps increasing.

Security endeavors currently employed:

A. Secure Communication Infrastructure

- We have seen cryptographic algorithms being designed and used to build secure networking protocols such as Internet Protocol Security (IPSEC), Transport Layer Security (TLS) or its predecessor Secure Sockets Layer (SSL), and Virtual Private Network (VPN).

B. Monitoring or Response System

- The research community has spent huge amount of effort to build a monitoring or response system. Firewalls, network-based IDSs, host-based IDSs and anti-virus programs have been widely deployed.

C. Built-In vs. Bolt-On Approaches in System Development

- In the Built-In approach, security features are designed up front and form part of the system development. The Bolt-On approaches compensate for the mistakes made earlier in the development cycle.

D. Code Instrumentation Tools and Self Checking Modules

- These techniques compute a flow graph using static or dynamic analysis, and instrument the program to check if the execution at runtime confirms to the flow graph.

Limitation of the current cyber security practices:

A. Secure Communication Infrastructure.

- If one endpoint is compromised , the crypto become helpless.

B. Monitoring or Response System.

- A perfectly safe monitor is yet to be designed

C. Built-In vs. Bolt-On Approaches in System Development.

- Bolt-On approach is the only solution for legacy systems.

D. Code Instrumentation Tools and Self Checking Modules

- Not generally effective against polymorphic exploits.

Towards a Holistic Security Approach

- Develop a comprehensive and modular software testing framework.
- Perform quantitative testing and analysis using game theoretic modeling.
- Enhance the quality of testing using knowledge management systems and self-testing software and hardware
- Make the system autonomous with provision to be controlled by the system administrator.

A Holistic Approach to Building Secure Systems

We propose a *Holistic Security Approach* which provides a framework that encompasses a whole system in a layered and organized manner.

For achieving optimal level of security, our approach collectively uses:

1. Monitoring tools
2. Knowledge base of attack patterns and solutions
3. Game theory inspired defense mechanisms

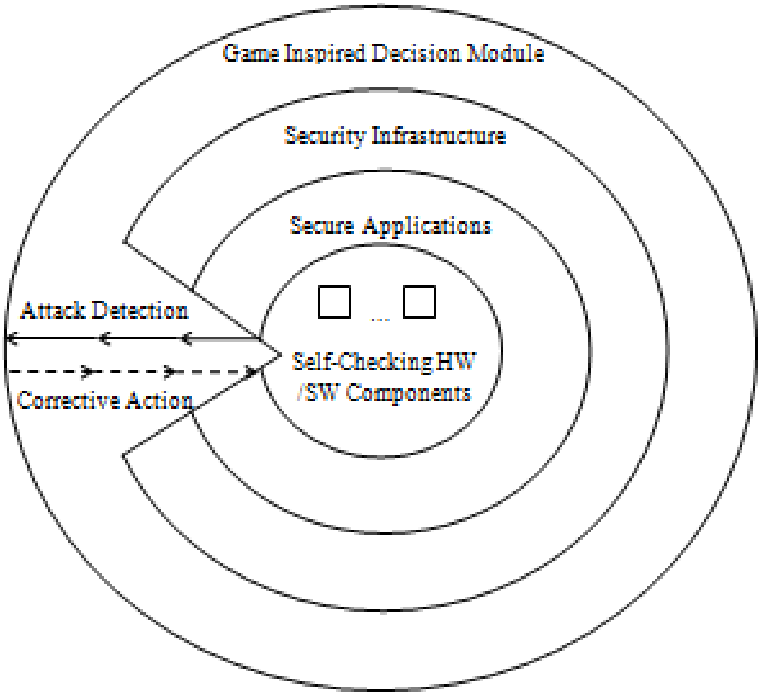
We emphasize on a carrot-and-stick approach for defense:

- We envision a security architecture where the defender plays a game with the attacker to observe his activities to further improve his defense strategies.

Four-layer Holistic Security Scheme

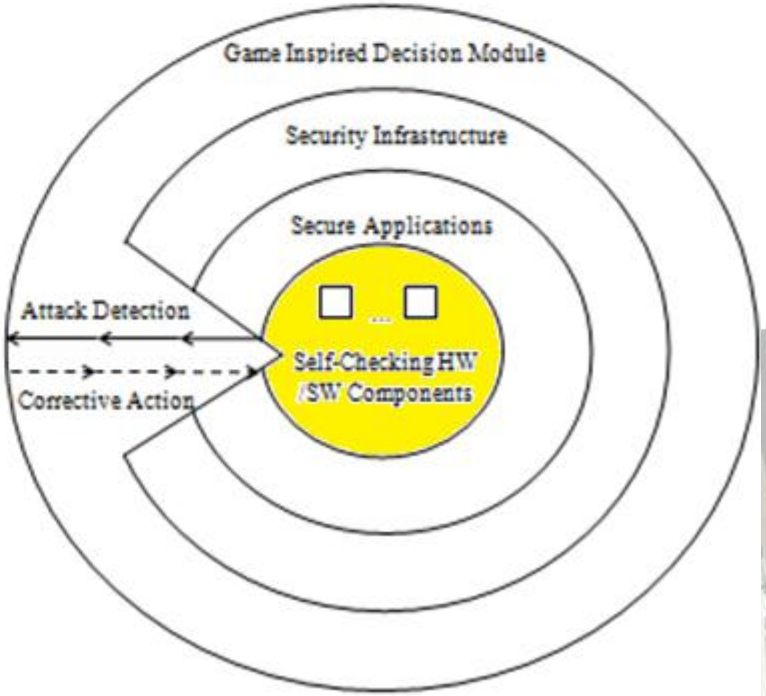
We envision a 4-layer Holistic Security Scheme:

1. Self-Checking HW /SW Components (Innermost layer *"The core"*).
2. Secure applications (Second layer).
3. Traditional network security infrastructur (3rd layer).
4. Game Theory Inspired Defense (Outer layer).



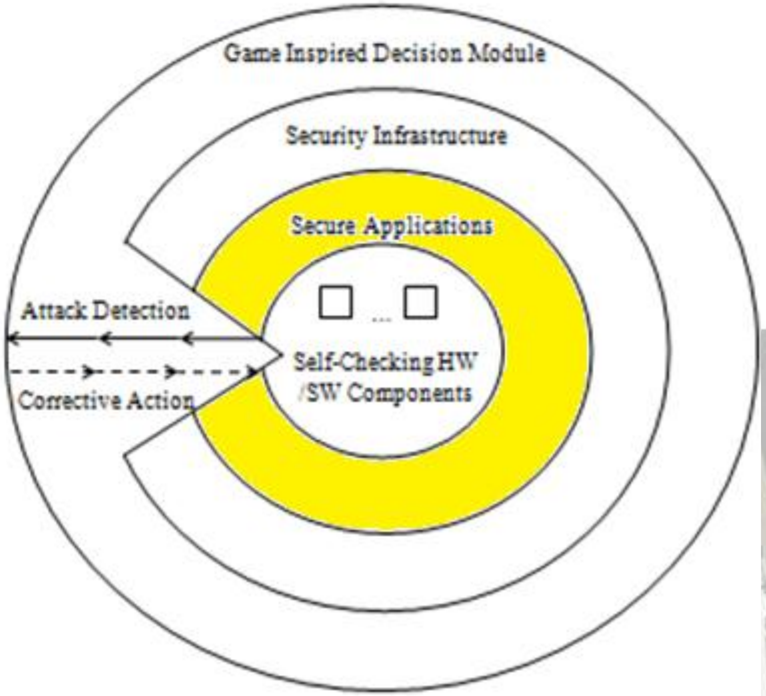
Self-Checking HW/SW Components (*“The core”*)

- Each HW/SW component has a provision of being wrapped with a self-checking feature.
- We assume the BIST methodology for monitoring hardware components.
- For monitoring software components we intend to use run-time monitoring tools to monitor security properties.



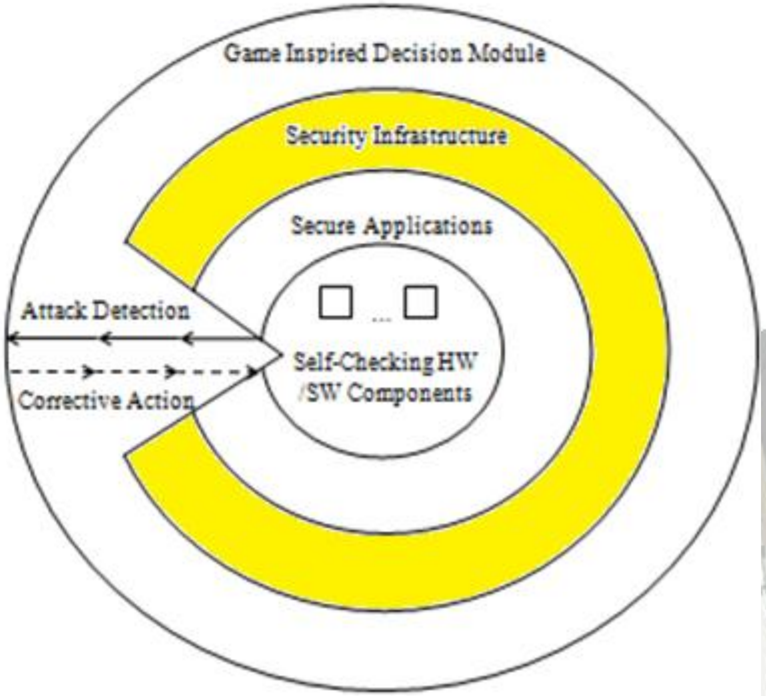
Secure applications (2nd layer)

- Consists of applications developed using built-in security methodologies; patched with enhancements as bolt-on.
- The *built-in* approach uses security requirements as part of the development methodology of the application being built.
- The *bolt-on* approach uses post-release patches and software updates to achieve the optimal level of security.



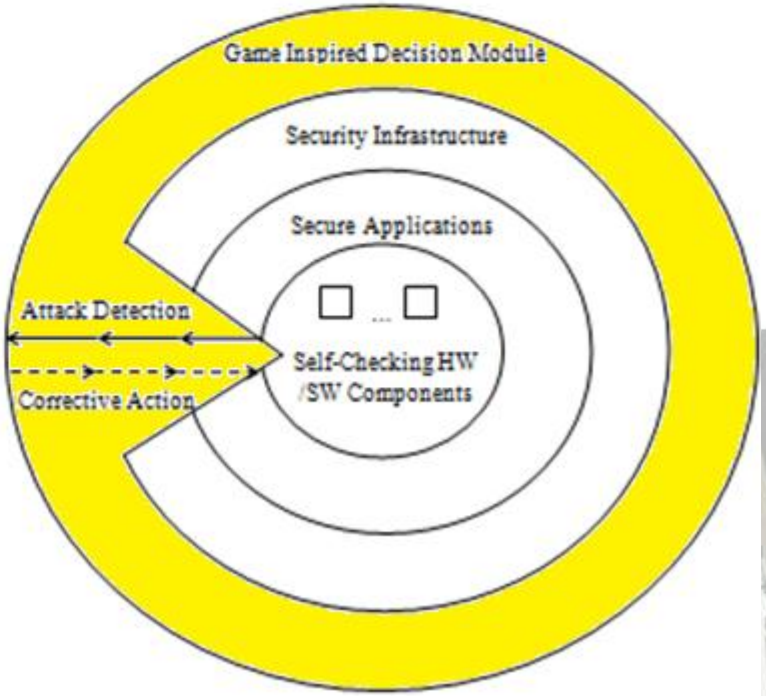
Traditional network security infrastructure (3rd layer)

- Implements techniques such as cryptographic algorithms.
- Primarily focuses on the use of tools such as: Intrusion Detection/Prevention Systems, Firewalls, Anti-virus/malware software.
- Provides protocols for communicating with the outermost layer which is the Game Inspired Decision Module.



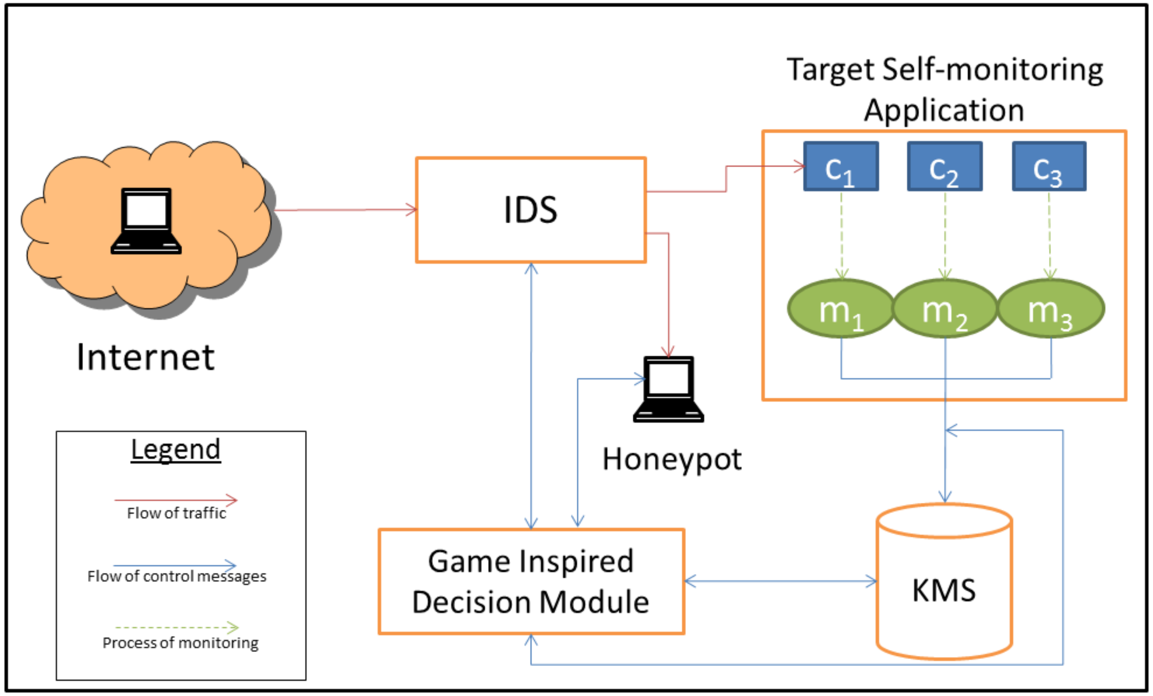
Game Inspired Decision Module (4th layer)

- Responsible for choosing the best security strategy for all the inner 3 layers
- Functions as the brain behind our holistic approach.
- GIDM and a Knowledge Management System accept inputs from the inner three layers.
- The KMS classifies the nature of an attack using attack vectors and associated defense measures.



Game Inspired Defense Architecture (GIDA)

- GIDA is an implementation of the described holistic approach for security testing.
- Emphasizes on testing the target system to find security breaches.
- Provides defense strategies against probable and committed attacks by modeling such situations as multi-player game scenarios.



Software Monitoring

- Monitors developed using monitoring tools will provide users flexibility to specify software properties to be monitored using logical formalisms.
- Monitors generated from formal specifications are then used to verify the execution of the program behavior.
- Once GIDM performs a decision analysis on the probability of attack, it informs the monitor of the appropriate action to take to minimize the damage to the application software.
- This information is also transferred to the KMS for appropriate attack classification.

Knowledge Management System

- Cyber attack taxonomy called AVOIDIT classifies attack vectors to assist defenders with disseminating defense strategies.
- Major classifiers are used to characterize the nature of an attack:
  - Classification by attack vector.
  - Classification by attack target.
  - Classification by operational impact.
  - Classification by informational impact.
  - Classification by defense.
- Cyber attack taxonomy is used as a repository schema for a Knowledge Management System (KMS).
- The KMS is used for regenerating the consummate path to an attack for propagating appropriate defenses.

Our Prior Work:

- We have computed the Nash Equilibrium strategy for a zero-sum stochastic game with imperfect information.
- We have computed Nash Equilibrium strategy for defending against DoS/DDoS attacks caused by UDP and TCP/TCP-friendly flows.
- We have devised AVOIDIT: A Cyber Attack Taxonomy
  - An enhanced attack taxonomy to accurately classify attack vectors at each stage of an attack, including blended attacks and disseminate defense strategies.

Future Work

- Test our proposed defense architecture on real world attack scenarios:
- Use attack test-beds to further investigate the efficiency of our holistic security approach.
- Research on evaluating and comparing potential defense game models, to maximize the defender's payoff.
- Investigate the application of Social Networking for enhancing cyber-security:
  - Concepts like: act of collusion among players, formal team formations, chain of trust, knowledge sharing, etc.

References

Shiva, S., Roy, S., Bedi, H., Dasgupta, D., and Wu, Q. “A Stochastic Game Model with Imperfect Information in Cyber Security”, The 5th International Conference on i-Warfare and Security, April 8-9, 2010.

Q. Wu, S. Shiva, S. Roy, C. Ellis, V. Datla, and D. Dasgupta. On Modeling and Simulation of Game Theory-based Defense Mechanisms against DoS and DDoS Attacks. 43rd Annual Simulation Symposium (ANSS10), part of the 2010 Spring Simulation MultiConference, April 11-15, 2010.

Bedi, H., Roy, S., Shiva, S., Game Theory-based Defense Mechanisms against DDoS Attacks on TCP/TCP-friendly Flows. IEEE Symposium on Computational Intelligence in Cyber Security (CICS), part of (SSCI). Paris, France. April 2011.

Chris Simmons, Sajjan Shiva, Dipankar Dasgupta, Qishi Wu, “AVOIDIT: A Cyber Attack Taxonomy,” IEEE Security and Privacy Magazine, under review.

Questions?