Stealth Attacks: Detection and Investigation

Ryan Jones + Thomas Mackenzie

Trustwave® SpiderLabs®

Ryan Jones

- Managing Consultant Incident Response at Trustwave SpiderLabs.
- Law Enforcement and Private Sector Experience
- Computer, mobile telephone and network investigations
- Lead Investigator for ‘Level 1’ Service Providers and Merchants

Thomas Mackenzie

- Application Security Consultant at Trustwave SpiderLabs
- Multiple Publications
- Co - Leader of OWASP Birmingham UK
- Founder of upSploit Advisory Management

Trustwave® SpiderLabs®

Agenda

- Part 1 – Developing a stealth attack
- Part 2 – Stopping the attacks at the framework

Part 1

Developing a stealth attack

Agenda

1. Blackhat vs. Investigator
2. Stealth Attack Principles
3. Stealth Attack – Live Demo
4. Stealth Attack Lessons
5. Stealth Attack Architecture Improvement

Blackhat vs. Investigator

- What do Blackhats do?
  - Steal information
  - Maintain access
  - Defacements
  - Common vulnerabilities
  - 0 days

Blackhat vs. Investigator

- What do Investigators do?
  - Gather data
  - Review logs
  - Recover data
  - Evaluate information
  - Draw conclusions

Blackhat vs. Investigator

- Tom = Blackhat
  - Find vulnerability
  - Exploit vulnerability
  - Upload web shell
  - Collect passwords

DEMO

Blackhat vs. Investigator

- Ryan = Investigator
  - Collect information
  - Web log analysis
  - Conclude what happened

DEMO

Blackhat vs. Investigator

- Debrief – Attack 1
  - IP address of attacker
  - Date and time of initial exploitation
  - Upload of malware
  - Malware actions
  - Date and time of password collection code

Blackhat vs. Investigator

- Introduction – Attack 2
  - Not happy!
  - What can be done to stop this logging?
  - More stealthy attacks…

Blackhat vs. Investigator

- Blackhat - What am I going to do differently?
  - Disassociate vulnerability finding IP from exploit IP
  - Use web shell not found by AV
  - Minimise usage of web shell
  - Remove the web shell after usage

DEMO

Blackhat vs. Investigator

- Ryan = Investigator
  - Collect information
  - Web log analysis
  - Conclude what happened

DEMO

Blackhat vs. Investigator

- Debrief – Attack 2
  - IP address of attacker
  - Date and time of initial exploitation
  - Upload of malware
  - Malware actions
  - Date and time of password collection code

Stealth Attack Principles

Elite hackers grant me the serenity to
accept the things that must be logged;
courage to change the things that don't;
and wisdom to know the difference.

Original by: Reinhold Niebuhr – Modified by: Ryan Jones

Stealth Attack Principles

1. Identify a vulnerability independently
2. Discover as much as possible legitimately
3. Analyse likely logging
4. Obfuscate accordingly
5. Remove trace

Stealth Attack

- Introduction – Stealth Attack
  - Take into account the five principles
  - Find the vulnerability
  - Exploit the vulnerability
  - Upload web shell
  - Collect password
  - Remove web shell
  - Sit back and relax…

RELAXING……

Stealth Attack Principles

1. Identify a vulnerability independently
2. Discover as much as possible legitimately
3. Analyse likely logging
4. Obfuscate accordingly
5. Remove trace

DEMO

Stealth Attack Principles

- Investigation
  - What evidence is available?
  - What evidence are we likely to find?

DEMO

Stealth Attack Principles

- Conclusion
  - Designing attacks around logs is highly achievable
  - Attackers gain assurance of less detection rates
  - Investigators lose assurance of detection rates

Part 2

Stopping the attacks at the framework

Agenda

1. Current application protection methods

2. Framework Level Security Profiling and Monitoring (FLSPM)

3. Demonstration of FLSPM

Application Protection Methods

- PHPIDS
  - Blacklisting with heuristic checks to find known attack signatures and variations
  - No profiling available, all or nothing
  - What risk level is okay?

Application Protection Methods

- WAF
  - Checks are made on requests using pattern matching and rule sets.
  - Checks are made on responses too, making sure nothing got past the input sanitisation.
  - Site wide profiling (normally)
  - No analysis on the application source code, rendering them “dumb” to what the application does.

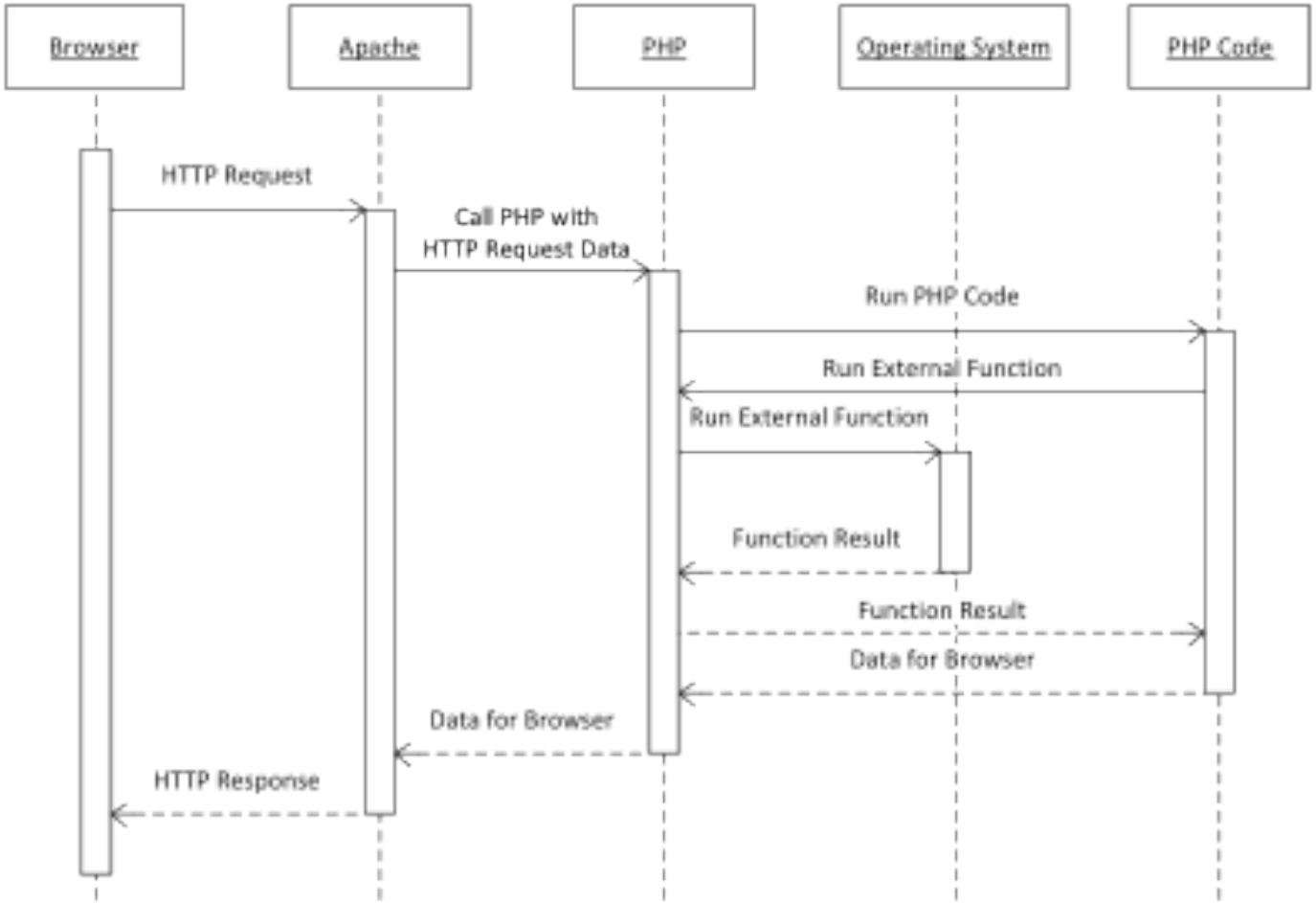
Application Protection Methods

- Hardened PHP -
  - Protects against vulnerabilities in the engine, runtime and session.
  - Filtering certain characters and requests.
  - Site wide profiling, no ability to profile for one particular page or separate applications.

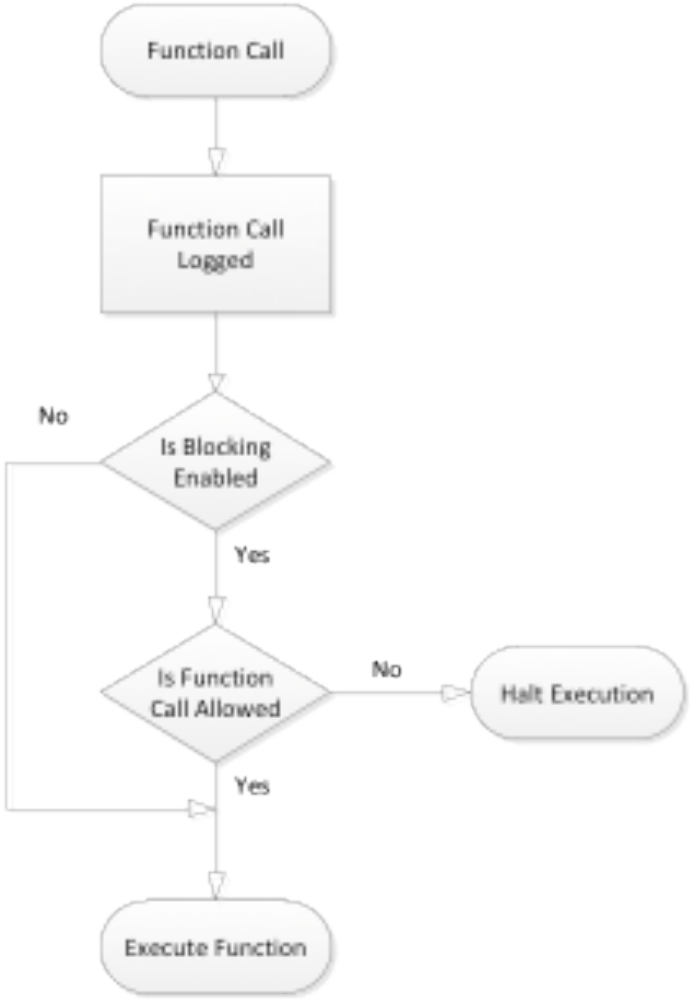
Framework Level Security Profiling and Monitoring

- Principles
  - Most systems rely on a single source of data.
  - Administrators have access to much more data.
  - Increase the detail of data → Reduces decision making complexity.

Framework Level Security Profiling and Monitoring



Framework Level Security Profiling and Monitoring



Framework Level Security Profiling and Monitoring

- Code
  - Logging
  - Blocking
  - Function Catch
  - Written in PHP

DEMO

Framework Level Security Profiling and Monitoring

- Limitations
  - PHP implementation: Trivial bypass
  - Profiling decisions are binary.
  - Limited data for decision making.

Questions?

ryan.jones@trustwave.com

tmackenzie@trustwave.com