Challenges of Automated Web Application Scanning

"Why automated scanning only solves half the problem."

Black Hat Federal
Washington D.C.
October 2, 2003
Speaker

Jeremiah Grossman (Chief Executive Officer)

- Founded WhiteHat Security in 2001
- Former Yahoo! Information Security Officer
- Performed over 400 web application assessments
- Primary developer of WhiteHat Arsenal, Web Server Fingerprinter, CIS Apache Benchmark Tool, and platform for Sentinel
- Frequent conference speaker at BlackHat, AFITC, Defcon, WSA, ISSA, ToorCon, etc.
- Whitepapers on Web Application Security, Cross-Site Scripting, and Cross-Site Tracing
- Credited on several major web application vulnerability advisories
Types of Security

- **Web Application Security**
  - Web Application Scanners
  - CGI Scanners
  - Web Application Firewalls

- **Network Security**
  - Firewall
  - Scanners (Qualys, ISS, Nessus)
  - Intrusion Detection

- **Host Security**
  - Intrusion Detection
  - Authentication
  - System Scanners
Web applications are vulnerable!

The Gartner Group

75% of the cyber attacks today are at the application level.

97% of the over 300 Web sites audited were found vulnerable to Web application attack.

_The WhiteHat team has witnessed similar results though its work experience._
Attractive targets

Web applications control the data that is most valuable.

- Credit Cards Numbers
- Bank Account Information
- Classified Information
- Personally Identifiable Information
- Medical History
- Personal Email

There is a web application for everything!
Compounding the problem

Conventional security solutions do not properly address the problem. Firewalls and SSL are not adequate security for a web application.

Frequent software updates and new web site functionality increase the potential for new web application vulnerabilities.

Web application security assessments require a tremendous amount of time, money, skill and diligence.
Automating vulnerability discovery

halting problem

The halting problem is a decision problem which can be informally stated as follows:

“Given a description of an algorithm and a description of its initial arguments, determine whether the algorithm, when executed with these arguments, ever halts.”

undecidable problem

“Not all problems can be solved. An undecidable problem is one that cannot be solved by any algorithm, even given unbounded time and memory.”
Humans vs. Scanners

Humans, as well as automated scanners, are best suited for identifying different types of security issues.

Scanners can be expected to be very thorough in their testing process and only identify easily identified “technical” vulnerabilities.

These automated scanners will not uncover multi-step procedure problems that often occur in complex web application. These procedural problems are referred to as “logical issues”.

A human possess the ability to analyze a large set of circumstances and determine, reasonably quickly, if a weakness in a process exists.
Logical vs. Technical

**Technical Flaws**

- Cross Site Scripting
- SQL Injection
- Directory Traversal
- Command Injection
- Frame Spoofing
- Buffer Overflows
- Directory Indexing
- Backup Files/Directories
- Configuration File Disclosure

**Logical Flaws**

- Manipulation of application business logic
- Price List Modification
- Account Privilege Expansion
- False Account Creation
- User Impersonation
- Unauthorized Funds Transfer

*Action requires a human intelligence.*
Technical Vulnerability

String of code or repeatable pattern that a computer can be programmed to recognize

If I put a single quote there and get an ODBC error then there is a SQL Injection vulnerability.

Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

[Microsoft][ODBC SQL Server Driver][SQL Server]Unclosed quotation mark before the character string "".

/products/shprodde.asp, line 129
“At step 3 of the wire transfer process, change the account parameter to point to the account you wish to transfer funds from. Continue changing the parameter on the next 2 steps of the transfer process.”
Logical Flaws in the News

Hackers Shortcut Hotmail
Password Reset Protections

According to information obtained by Newsbytes, hackers recently discovered a way to skip the validation form and go directly to any user's "secret question" prompt. From there, the intruder is only one step away from resetting the user's password. Sources say that since the discovery of the security hole roughly two weeks ago, a small cadre of hackers has been patiently checking a long list of high-profile and desirable usernames for easily-guessed answers to secret questions.

http://www.computeruser.com/news/02/02/13/news2.html
We need a solution that makes sense!

“If a scanner alone will not complete the job by itself, then a combination of software and security personnel is required.”

- Identify all technical and logical security issues.
- Be able to handle large web sites.
- Be able to maintain a logged-in state.
- Low volume of false positives
- Scheduled
- Consistently current
- Ability to scan remotely with no source code access
Using and Building Scanners

For years we tested all forms of free and commercial web application scanning tools and utilities as consultants, developers and administrators.

* Disappointed in all available solutions *

WhiteHat formed a team of industry leading web application security professionals, web application developers, and statistical analysis engineers.

WhiteHat’s team spent the last two years developing the latest in Web application scanning technology.

In the process of developing web application scanners, a tremendous amount of R&D was required to handle unforeseen challenges.
Remote Testing

Automated web applications scanners use a remote black box approach. All web applications are different. Different software, platforms, and configuration.

Network Security Scanning:
“Identifying known vulnerabilities in known code.”

Web Application Security:
“Identifying known classes of vulnerabilities in unknown code.”
Automated Scanning Challenges

Automated Login
Logout Detection
Infinite Web Sites
Authentication System Auditing
Errors and Responses
Multi-Step Processes
Strange URL Structure
Client-Side Generated Links
Automated Login

The web application scanner must be able to generically login to a web application on demand.

**Sign In**

- **Online ID:**
  
  (6 - 20 numbers and/or letters)

- **Passcode:**
  
  (4 - 8 numbers and/or letters)

- Remember my online ID ([How does this work?])

- Sign In

[Forgot your Passcode? Create your Passcode] [Forget your ID?]

A scan is largely invalid if scanned while not properly authenticated because full functionality cannot be exercised.
Automated Login

The login process must support:

- Cookie and URL based session credentials.
- Multiple pieces of authentication information, beyond the simple username and password.
- Multiple re-directs.
- SSL
Automated Login

The login process must support:

• Client-Side Scripting Languages.

<script language="javascript">
/*
 * A JavaScript implementation of the RSA Data Security, Inc. MD5 Message Digest Algorithm, as defined in RFC 1321.
 * Copyright (C) Paul Johnston 1999 - 2000.
 * See http://pajhome.org.uk/site/legal.html for details.
 */
</script>
A scanner will at some point become logged out. How does the scanner know when it's been logged out?

Logout occurs by:
- Clicking logout links
- Timing out
- Application errors
- Session expiration
- etc, etc, etc,
Detecting Logout

We used a system that performs preliminary tests on the web application to learn the login/logout nuances.
Infinite Web Sites

The website is enormous and crawling the entire site in a reasonable amount of time is impossible. Must compile an accurate structural map.

Dynamic Web Sites:
• Rate of addition
• Rate of decay
• Very large database of items 500,000+ links
• Dynamic URL creation
Infinite Web Sites

Condense the amount of links we need to crawl and create a complete structural map of the site.

Locate:
All web applications
All unique parameter name instances
Many web application authentication systems are inherently weak. They can be susceptible to session hi-jacking, session replay, etc.

Cookie: T=user=admin

Or

Cookie: S=UID=ae5fad5ad6a8asd6as9

Even if the scanner could twiddle the bits, how does scanner know when something works or does not work or what's good or what's bad? How does a scanner know when it accesses another bank account? "Scanner is not able to generically determine context of good or bad"
Response Codes and Errors

Not Found does not always mean, “Not Found”.

- Not everyone is RFC compliant
- Universal Error Catching
- Error strings are different
Response Codes and Errors

Error: Not Found

We think you are lost...
There is no such URI on this server!

Tyker Records®
Response Strings

Application Errors

SQL Injections

XSS

Command Injection

Removing response messages helps prevent against exploitation. However, prevents scanners from finding the vulnerabilities. Lots of false positives.
Multi-Step Process

Create a User ID and Password

The password you have created is either too short or too long. Please create a password that is 6 - 32 characters in length. Thank you.

First Name: Jeremiah
MI: J
Last Name: Grossman

E-mail Address: jeremiah@whitehatsec.com

Create a User ID*
jeremiah
6 - 32 alphanumeric characters. (Do not create a User ID containing 9, 16 or 17 numbers such as your social security number or ATM card number).

Create a Password* Confirm Your Password*
******* *********
6 - 32 alphanumeric characters

*Your User ID and Password are case sensitive, so remember if you use upper and lower case letters.

My Password Hint
HINT
Enter a hint to remind you of your Washington Mutual Password. For example "My favorite color" or "My dog's name".
If a computer is not supposed to automate this process, then how can a scanner?
Strange URL Structure

There are some very strange looking URLs these days. The normal web application url structure has a “?” delimiting the file name from the parameters. However, developers have realized that many web spiders will not index dynamic data so they have opted for some non-standard trickery.

The goal is to identify:
• Web application filename
• Web application parameter names and values

Even if:
• There is no question mark
• No “&” and uses strange delimiters.
• Strange file extension (like .html)
Normal URL Structure

Normal:

/articles/03/08/19/1748206.shtml?tid=109&tid=111&tid=126
/news?hl=en&edition=us&q=a&btnG=Search+News
/shopping/category.asp?categoryID=11
/weeknight_survival.asp?wday=3&ww=this

Inject into the name value pairs.
Strange URL Structure

Strange:

/gp/browse.html/10217298046144934?node=1036592

/exec/obidos/ASIN/B00009J5VW/ref=e_hp_cb_3_1/12-1729804-6144934

/srs7/sid=030803095821064050032/g=home/search/detail/base_pid/271134/

/catindex/computers.html?ssPageName=MOPS5:HEC03

/exec/obidos/subst/home/home.html/102-17298046144934

/shop/enter.asp?category=2378467~2378483
Client Side Generated Links

Sometimes websites will have menus and style sheets which create hyperlinks on the fly. In these cases, web crawlers have a extremely difficult time traversing the site since the links are not yet built and parseable.

“Unsolved problem by all web crawlers.”
Web applications can be extremely fragile, especially where there is database access.

*Run the scans low and slow.*
What have we learned?

“A web application scanner can alleviate a tremendous workload in a penetration test. However, software alone cannot be expected perform the entire task of securing a web application”.

“All web application scanners find vulnerabilities using error messages. If error messages are suppressed, vulnerabilities are exponentially harder to detect using automated means.”

“All web application scanners will produce a high volume of false positives.”
Humans and Scanners

“Human assessments and scanners are required for complete vulnerability coverage when it comes to web applications.”

**Confidential Information Disclosure**
- Verbose Error Messages
- HTML Comments
- Known Directory
- Known CGI File
- Configuration File Disclosure
- Backup File Disclosure

**Application Input Manipulation**
- SQL Injection
- Cross-Site/In-Line Scripting
- Buffer Overflow
- OS Command Injection
- Meta Character Injection
- Directory Traversal
- Null Injection
- User-Agent Manipulation
- Referrer Manipulation
- Debug Commands
- Extension Manipulation
- Frame Spoofing

**Session Management**
- Brute/Reverse Force
- Session Hi-Jacking
- Session Replay
- Session Forging
- Password Recovery

**Logical Vulnerabilities**
- Logical Flaws (Manipulation of application business logic)
- Account Privilege Escalation
- Page Sequencing
- User Impersonation
- Improper Session Handling
Thank You - Questions?

Jeremiah Grossman
Jeremiah@whitehatsec.com
http://www.whitehatsec.com