Automated Hacking via Google

- By Daniel Bartlett, 21c3 December 2004

freebox Security and Development Team

- A small group of people spread over the world
- Enjoying research and development
- Relish a discussion
- Involved in a few open source projects

Outline

- Google – The info search engines provide
- PHP – The language
- The Issues – Bad coding and what it causes
- Inclusion – The possibilties
- Automation – Speed up the whole process
- PHP Worm – The ideas behind the code
- Analysis of the Santa worm

Search Engines

- Any search engine provides well structured HTML or XML based search results
- Provides a quick method of finding other web servers
- Allows data collation without communication with target
- Search for known vulnerablity, search for errors

The PHP Language

- Easy to learn
- Lots of resources to learn from
- Many Open Source classes/functions to utilise
- Aids in Rapid Development

Security Issues

- **Local File Inclusion** – Information Disclosure, execution of uploaded scripts, execution of scripts in the incorrect context
- **Remote File Inclusion** – Execution of foreign scripts, probably the most dangerous
- **SQL Injection** - Information Disclosure; any data in the database that the PHP script has access to, unless they used the root account; all! And updating or insertion of new data
- **File Upload** – Overwriting of content, can be used in conjunction with Local File Inclusion

Simple Protection Functions

```
function white_list($indata) {
    $white = array('home', 'products', 'contact');
    if(in_array($indata, $white)) return $indata;
    else return “”;
}



function black_list($indata) {
    $black = array('http', 'ftp', 'union', '..', '\');
    for_each($black as $value) {
        $indata = str_replace($value, “”, $indata);
    }
    return $indata;
}
```

Remote file Inclusion

We developed an includable file containing the following functionality:

- Browsing the file system
- Viewing, Editing and Uploading of files
- A sudo command line
- Browsing databases(MySQL/ODBC)
- TCP Port scanning
- Sending MIME emails
- Installation of C based Connect Back/Listining shell
- Debugging of the Script and Global variables

Automation

Start it simple. Google for known vulnerabilities then test each result.

Expand by looking for unknown holes, starting with Error Codes from PHP; like “Failed opening for inclusion”, “Undefined variable”, etc. Then test each result.

Walking of pages by grabbing the page and looking for links and testing each one with a “fuzz” set of common variables, or looking at the variables used in the site and then bruting them.

The most rewarding is manually walking a site trying each variable you come accross, takes a long time but I get very pleased when a site becomes a site with a hole.

PHP Worm

- Portable code – Runs on any server
- Short execution time – Maximise number of executions
- Infection vectors – Many routes for attack
- Target discovery – Search Engines, Subnet Probing, TCP Port Scanning
- Mutation – Safer transportation
- Peer To Peer – Build a web net, no single point of failure

Analysis of Santa.a Worm

- Has only one infection vector
- Weakness in requiring Google
- Coded in Perl, limiting target hosts
- Code didn't always transfer sucessfully
- Defaces sites rather than building a network

Web Based Worm

- Is multi language – PHP/Perl/ASP/Bash/etc.
- Knows multiple know vulnerability
- Searches for error messages from all lang's
- Mutates on each infection