**PART B**

|  |
| --- |
| **Automated Checks** |

This is one out of many examples for a simple Perl scripts that scans a IIS webserver for existing files/directories (contained in the input file iis.txt).

use LWP::UserAgent;

my $url = "212.41.205.98";

open (DICT, "iis.txt");

my @lines = <DICT>;

chomp @lines;

foreach $line (@lines)

{my $request = new LWP::UserAgent;

my $output = $request->get("http://$url$line");

if($output->is\_success())

{print "url found: $line\n";}}

Following is the .Net code that will execute and get the results from the automated checks. Highlighted portions are what the .net platform will be interested in.

ProcessStartInfo perlStartInfo = new ProcessStartInfo(@"d:\Perl\bin\perl.exe");  
perlStartInfo.Arguments = "c:\\simple\_urlscan.pl " + "c:\\randomefilename.txt";  
perlStartInfo.UseShellExecute = false;  
perlStartInfo.RedirectStandardOutput = true;  
perlStartInfo.RedirectStandardError = true;  
perlStartInfo.CreateNoWindow = false;  
Process perl = new Process();  
perl.StartInfo = perlStartInfo;  
perl.Start();  
perl.WaitForExit();  
string output = perl.StandardOutput.ReadToEnd();

**Explanation**

For the simple\_urlscan.pl the input is a list of URLs.

* On the web I provide an input field for it.
* Then I write a .net wrapper class that will create the input file.
* Then I use the above code to execute the script and read the output.

You as a Perl developer will **ideally** not require interacting with the data base at all. What you **would do is to dictate the input file data structure** (text file format) so the web developer could generate it.

**Rules of perl script**

* You MUST NOT HARDCODE THE FILE NAME ,
* You Must take the full file path as a parameter.
* You can create a temporary working file for the purpose of the script
* The output must be return to the STDOUT. (so the web developer can read it via the .net code above)
* If the Perl script produces a trace image, the output should send a list of image files that the web developer created.

**More scripts**

Below you find a lot of ready to use Perl scripts or URL’s that will help get the information All is optional – if you feel you can program it yourself faster: give it a try!). They all need to be adapted so the input variables will be delivered via the frontend and the output is parsed into the DB and finally is also visible within the frontend. The following list covers about ¾ of the automated checks. A few more will come.

| **Test Category** | **Infos to collect/Question** | **Tool/Link** |
| --- | --- | --- |
| DNS | Get the nameservers | (or <http://ha.ckers.org/fierce/> also is a nice perl doing some of the stuff) |
| DNS | get the MX record |
| DNS | Perform axfr queries on nameservers |
| DNS | Get extra names via google scraping |
| DNS | Brute force subdomains from file   1. Option 1. Small txt input file 2. Option 2: big txt input file |
| DNS | WHOIS |  |
| DNS | Typosquatting | <http://serversniff.net/typosquatting.php> |
| DNS | Find Top level domains |  |
| DNS | Find hostnames on a IP | <http://serversniff.net/content.php?do=hostonip> |
| DNS | What domains can be found using google? | <http://code.google.com/p/dnschk/> |
| DNS | SOA (State of Authority) record |  |
| DNS | SOA (State of Authority); serial numbers |  |
| DNS | Is there an a-record for the domain? |  |
| DNS | Fingerprint | http://code.google.com/p/dnschk/ |
| DNS | Check DNS servers against the kaminsky | http://code.google.com/p/dnschk/ |
| DNS | Is there a SPF record? |  |
| DNS | Get nameserver version | http://code.google.com/p/dnschk/ |
| DNS | Get reverse DNS records for ip ranges |  |
| WEB | What URL?s can be extracted from a website? | <http://code.google.com/p/urlgrep/> |
| Information Gathering | What results can be collected from a fast portscan? | <http://pentestmonkey.net/yaptest/overview/yaptest-overview> |
| IG | What ports (UDP/TCP) can be detected? | <http://code.google.com/p/nmap-parser/> |
| Web | What Webvulnerabilities can be detected with nikto? | <http://cirt.net/nikto2> |
| Web | What CMS can be detected? | <http://code.google.com/p/cms-explorer/>  <http://code.google.com/p/web-sorrow/> |
| Web | Is Webdav enabled and can it be exploited? | <http://code.google.com/p/davtest/> |
| Web | How is the SSL quality of the target site? | <http://code.google.com/p/sslaudit/> |
| Web | Can login pages be detected? | <http://code.google.com/p/web-sorrow/>  <http://packetstormsecurity.org/files/92222/Admin-Control-Panel-Finder-2.0.html> |
| Web | What error messages can be generated? | <http://code.google.com/p/web-sorrow/> |
| Web | Can any interesting file be spottet? | <http://code.google.com/p/web-sorrow/> |
| Web | Is the Apache server vulnerable to a Range Header Dos? | <http://code>.google.com/p/apache-range-header-test/ |
| Web | SQL injection possible? | <http://code.google.com/p/bsqlbf-v2/>  <http://code.google.com/p/firefuzzer/>  <http://code.google.com/p/skipfish/> |
| Web | XSS possible? | <http://code.google.com/p/firefuzzer/>  <http://code>.google.com/p/skipfish/ |
| Web | Is there a vulnerability in the Joomla CMS? | <http://packetstormsecurity.org/files/106444/Joomla-Web-Scanner-1.7.html> |
| Web | Fuzzing: what results are generated with a fuzzer? | <http://packetstormsecurity.org/files/104872/Short-Fuzzy-Rat-Scanner.html> |
| Web | What information can be read from the HTTP banner? | http://packetstormsecurity.org/files/104569/Plomp-Banner-Grabber.html |
| Web | Is it possible to exploit a vulnerability in AWSTATS? | <http://packetstormsecurity.org/files/82352/AWStats-migrate-Remote-Command-Execution.html> |
| SMTP | Relay/Spoofing possible? | <http://packetstormsecurity.org/files/36831/relayscanner.zip.html> |
| FTP | Can existing FTP account be cracked? | <http://packetstormsecurity.org/files/100647/SSH2FTPCrack-FTP-SSH-Brute-Forcer-0.2.html> |
| SSH | Can existing SSH account be cracked? | <http://packetstormsecurity.org/files/100647/SSH2FTPCrack-FTP-SSH-Brute-Forcer-0.2.htm> |
| DoS | How is the host reacting when flooded with UDP packets? | <http://packetstormsecurity.org/files/24362/udp.pl.html> |