CSE 565 - Homework 7

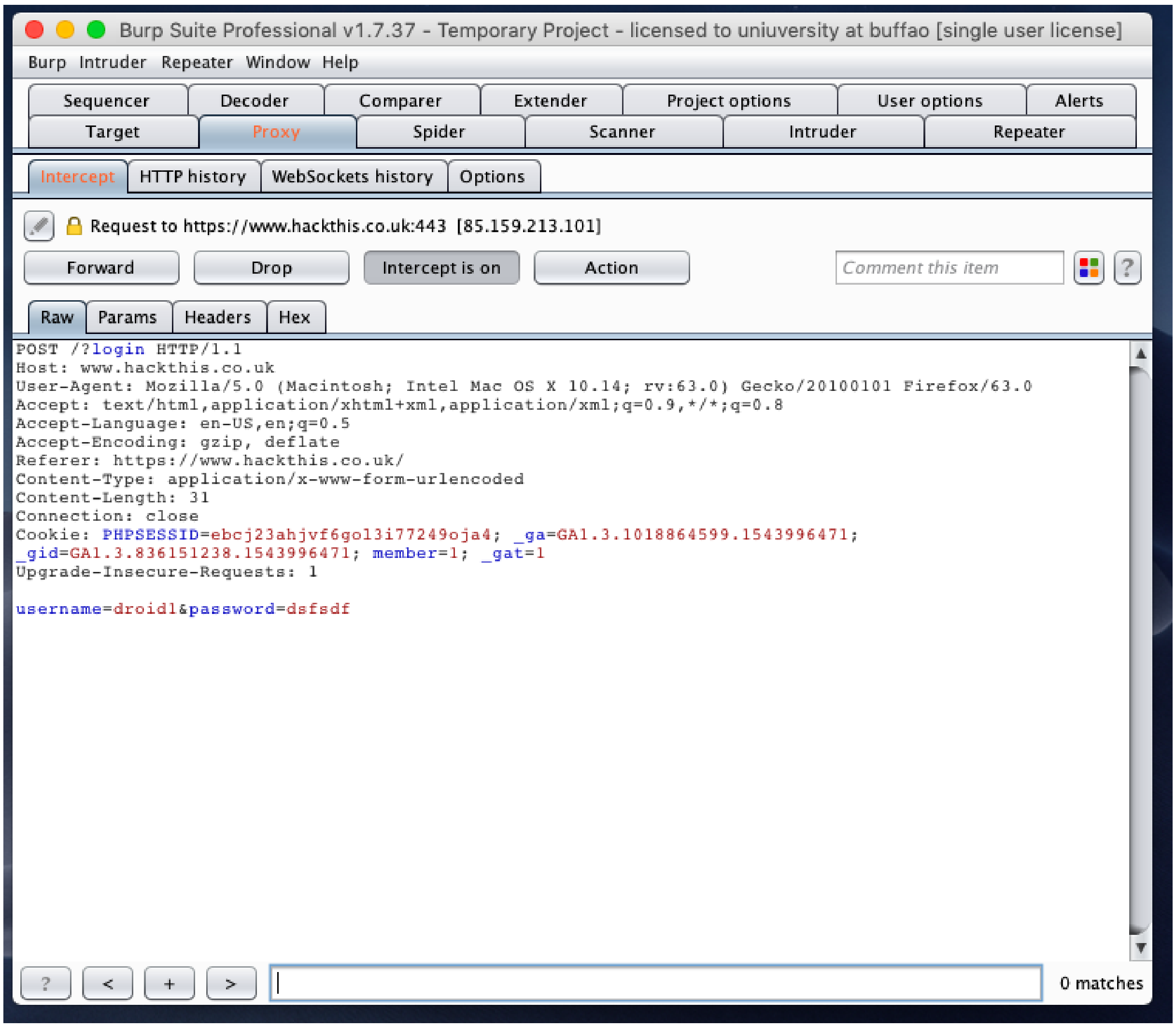
Amlan Gupta (#50288686)

December 2018

1 Question

Identify a suitable fuzzing tool for a system that you know (this can be one of the tools mentioned in the lectures). Run the tool you choose on a reasonably large program (e.g., an open source application), or a trace produced by it, perform testing, and report your findings. Also comment on the ease of using this tool.

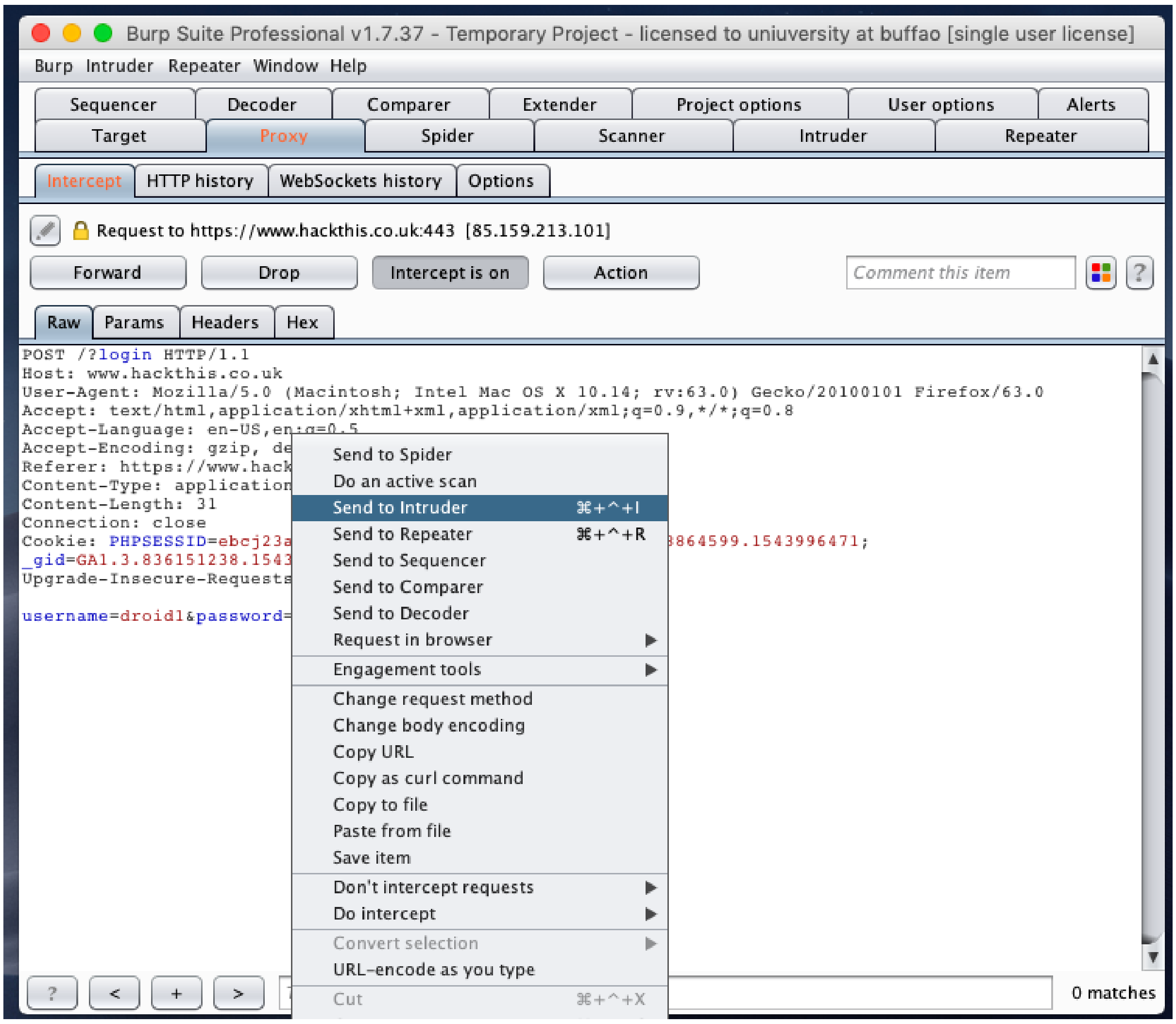
We will be using Burp Suite which is a graphical tool for testing Web application security. The testing will be performed on https://www.hackthis.co.uk [1] website which lets us perform unlimited number of login attempts. An user account was created with insecure passord and performing brute-force technique we will be trying to get unauthorized access to user account information.


Burp Suite Professional v1.7.37 - Temporary Project - licensed to uniuversity at buffao [single user license]
Burp Intruder Repeater Window Help
Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater
Intercept HTTP history WebSockets history Options
Request to https://www.hackthis.co.uk:443 [85.159.213.101]
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
0 matches

```
POST /?login HTTP/1.1
Host: www.hackthis.co.uk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.hackthis.co.uk/
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Connection: close
Cookie: PHPSESSID=ebcj23ahjvf6gol3i77249oja4; _ga=GA1.3.1018864599.1543996471;
_gid=GA1.3.836151238.1543996471; member=1; _gat=1
Upgrade-Insecure-Requests: 1

username=droid1&password=dsfsdf
```

Figure 1: Intercepting HTTP request

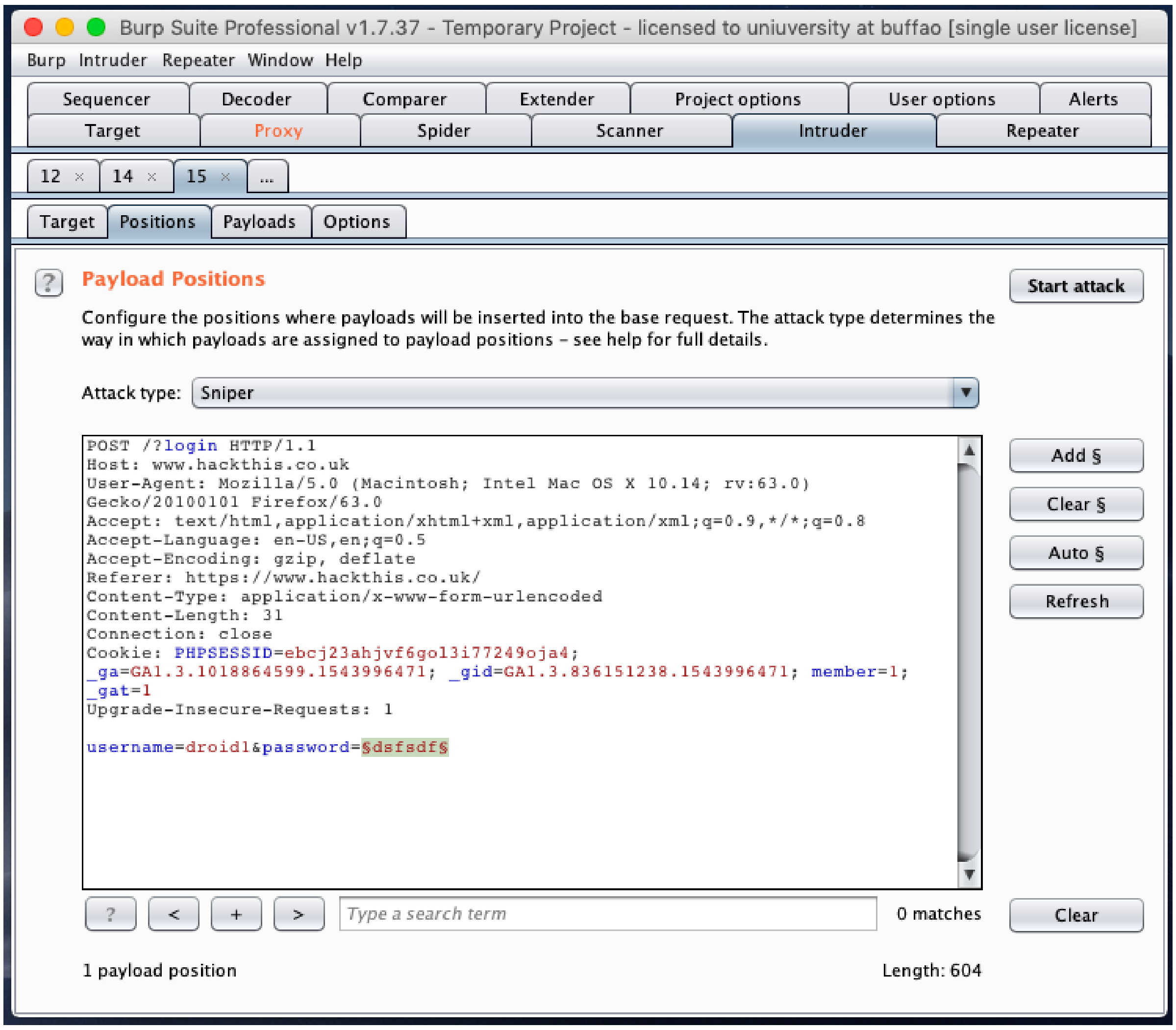
We try to make a login attempt with on the website using Firefox, however with the help of burp suite we intercept the request. We will be using this request template as a framework to perform unauthorized login.


Burp Suite Professional v1.7.37 - Temporary Project - licensed to uniuversity at buffao [single user license]
Burp Intruder Repeater Window Help
Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater
Intercept HTTP history WebSockets history Options
Request to https://www.hackthis.co.uk:443 [85.159.213.101]
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
Send to Spider
Do an active scan
Send to Intruder ⌘+^+I
Send to Repeater ⌘+^+R
Send to Sequencer
Send to Comparer
Send to Decoder
Request in browser
Engagement tools
Change request method
Change body encoding
Copy URL
Copy as curl command
Copy to file
Paste from file
Save item
Don't intercept requests
Do intercept
Convert selection
URL-encode as you type
Cut ⌘+^+X
? < + >
0 matches

```
POST /?login HTTP/1.1
Host: www.hackthis.co.uk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, de
Referer: https://www.hack
Content-Type: application
Content-Length: 31
Connection: close
Cookie: PHPSESSID=ebcj23a                                 3864599.1543996471;
_gid=GA1.3.836151238.1543
Upgrade-Insecure-Requests

username=droid1&password=
```

Figure 2: Forge similar request

We send the request to 'Intruder', which will forge a similar request to replicate a valid login attempt.


Burp Suite Professional v1.7.37 - Temporary Project - licensed to uniuversity at buffao [single user license]
Burp Intruder Repeater Window Help
Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater
12 × 14 × 15 × ...
Target Positions Payloads Options
Start attack
Add §
Clear §
Auto §
Refresh
? < + >
Type a search term
0 matches
Clear

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions – see help for full details.

Attack type: Sniper

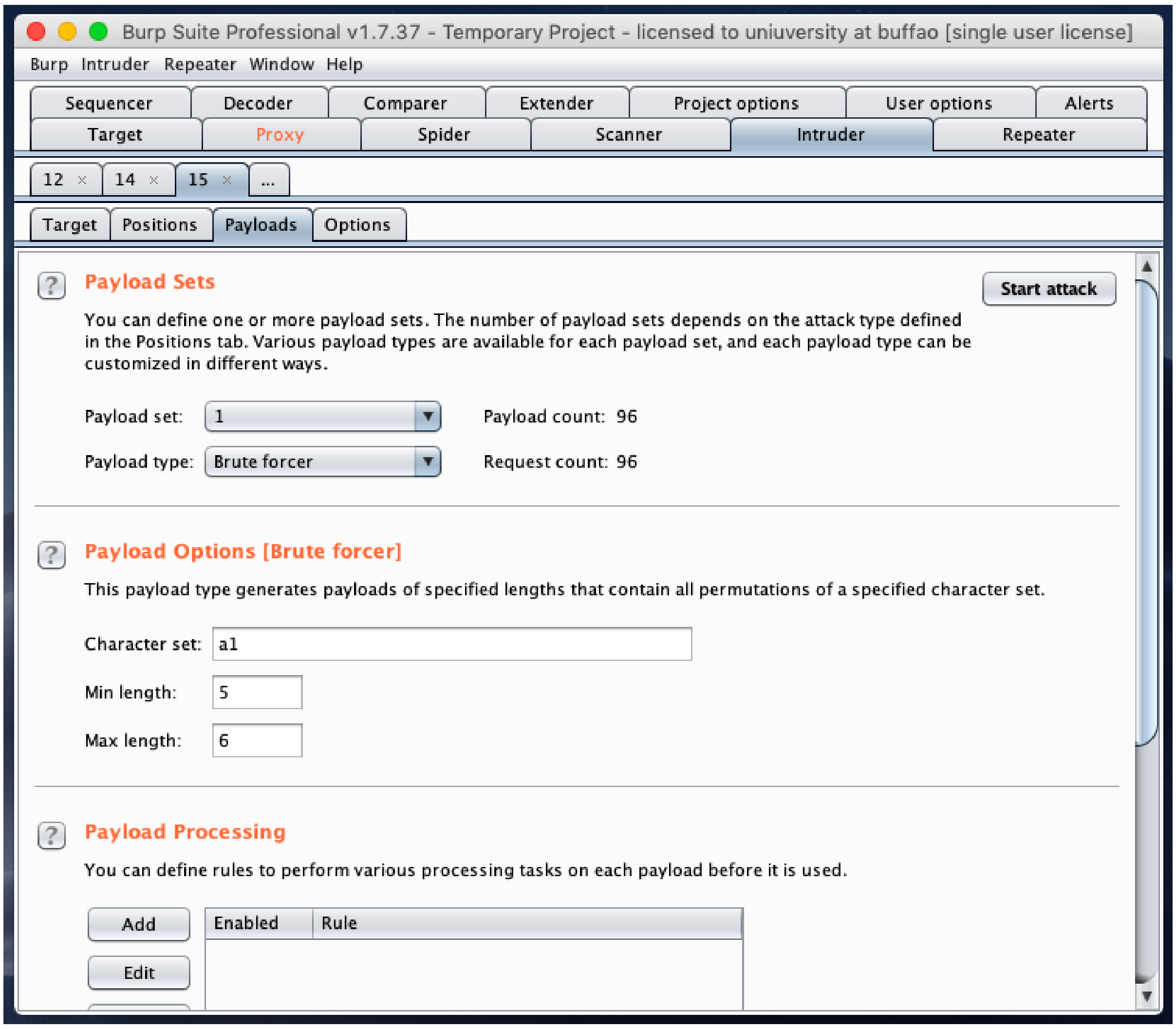
```
POST /?login HTTP/1.1
Host: www.hackthis.co.uk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:63.0)
Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.hackthis.co.uk/
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Connection: close
Cookie: PHPSESSID=ebcj23ahjvf6go13i77249oja4;
_ga=GA1.3.1018864599.1543996471; _gid=GA1.3.836151238.1543996471; member=1;
_gat=1
Upgrade-Insecure-Requests: 1

username=droid1&password=§dsfsdf§
```

1 payload position

Length: 604

Figure 3: Set variable

Keeping all the variables in the request constant we will be trying with different password, hence we set the value of password field as the variable.


Burp Suite Professional v1.7.37 - Temporary Project - licensed to uniuversity at buffao [single user license]
Burp Intruder Repeater Window Help
Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater
12 14 15 ...
Target Positions Payloads Options
Start attack
Add
Edit
Enabled Rule

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 96

Payload type: Brute forcer Request count: 96

Payload Options [Brute forcer]

This payload type generates payloads of specified lengths that contain all permutations of a specified character set.

Character set: a1

Min length: 5

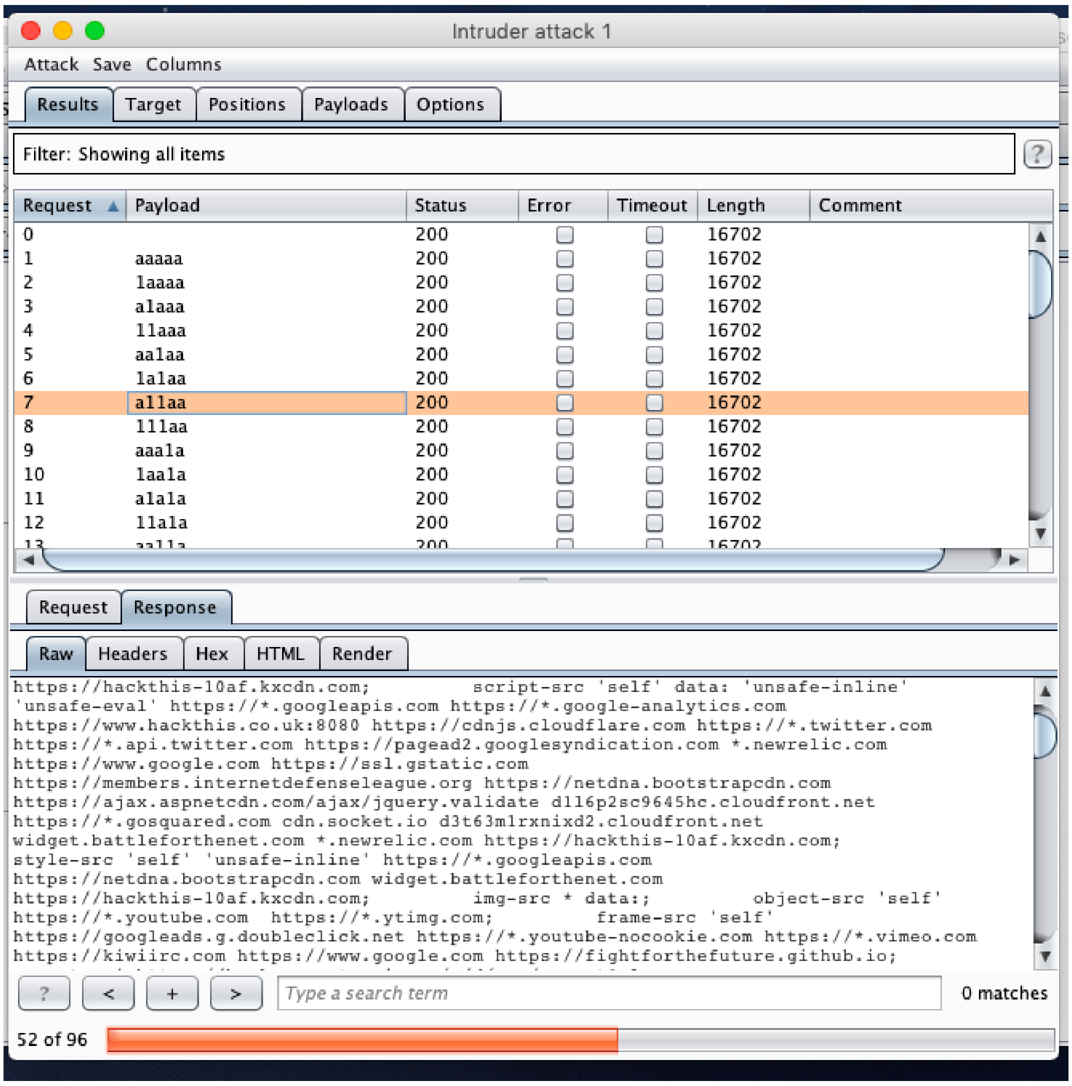
Max length: 6

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Figure 4: Payload creation

We will try a brute-force approach to perform a successful login attempt. For simplicity's sake, we are trying with 5 to 6 characters length strings consists of a and 1. In real life situation, all the characters needs to be included which will create millions of combinations. However we can see, in this case we will be performing 96 attempts.

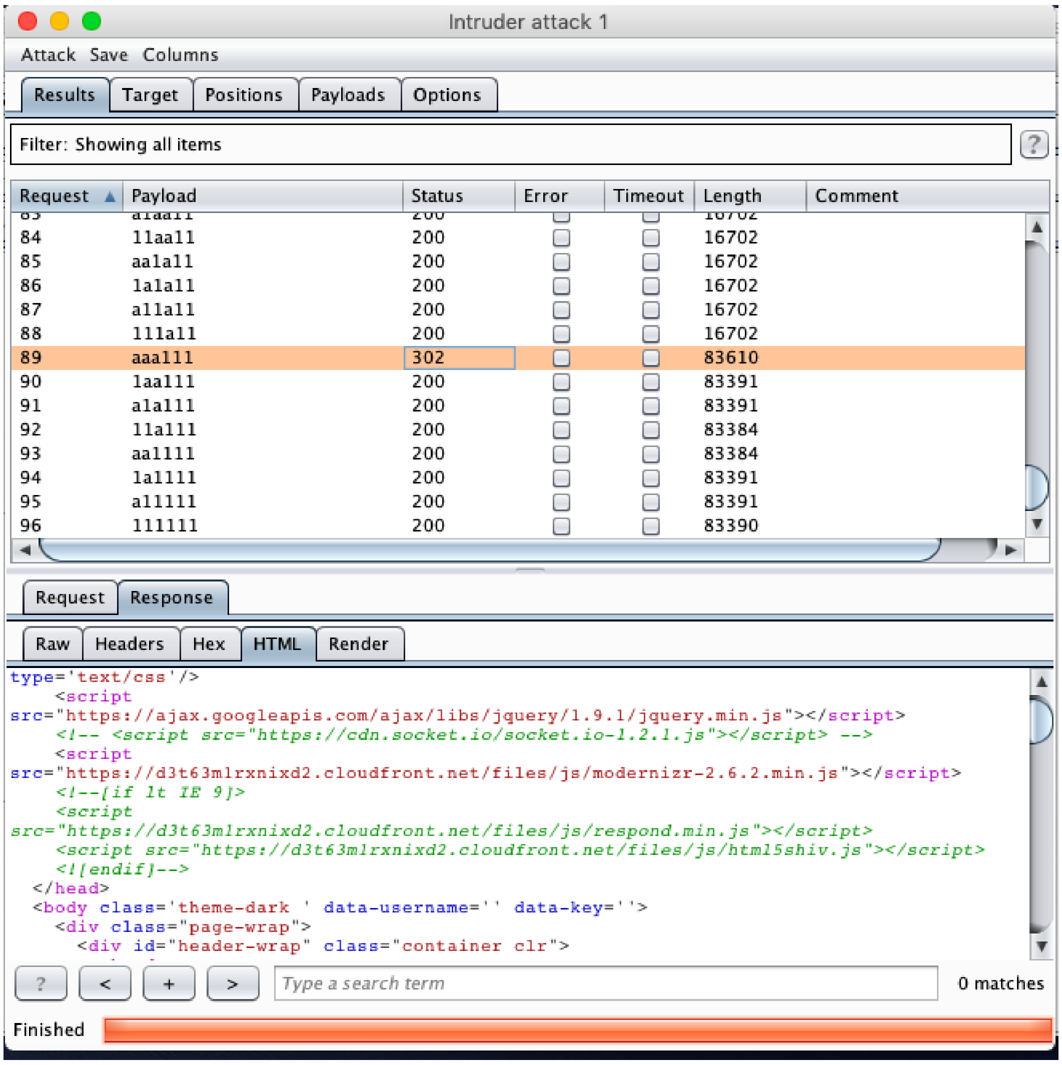

Intruder attack 1
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Response
Raw Headers Hex HTML Render
Type a search term
0 matches
52 of 96

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	16702	
1	aaaaa	200	☐	☐	16702	
2	laaaa	200	☐	☐	16702	
3	alaaa	200	☐	☐	16702	
4	llaaa	200	☐	☐	16702	
5	aalaa	200	☐	☐	16702	
6	lalaa	200	☐	☐	16702	
7	allaa	200	☐	☐	16702	
8	lllaa	200	☐	☐	16702	
9	aaala	200	☐	☐	16702	
10	laala	200	☐	☐	16702	
11	alala	200	☐	☐	16702	
12	llala	200	☐	☐	16702	
13	aalla	200	☐	☐	16702	

```
https://hackthis-10af.kxcdn.com;         script-src 'self' data: 'unsafe-inline'
'unsafe-eval' https://*.googleapis.com https://*.google-analytics.com
https://www.hackthis.co.uk:8080 https://cdnjs.cloudflare.com https://*.twitter.com
https://*.api.twitter.com https://pagead2.googlesyndication.com *.newrelic.com
https://www.google.com https://ssl.gstatic.com
https://members.internetdefenseleague.org https://netdna.bootstrapcdn.com
https://ajax.aspnetcdn.com/ajax/jquery.validate d116p2sc9645hc.cloudfront.net
https://*.gosquared.com cdn.socket.io d3t63m1rxnixd2.cloudfront.net
widget.battleforthenet.com *.newrelic.com https://hackthis-10af.kxcdn.com;
style-src 'self' 'unsafe-inline' https://*.googleapis.com
https://netdna.bootstrapcdn.com widget.battleforthenet.com
https://hackthis-10af.kxcdn.com;         img-src * data:;         object-src 'self'
https://*.youtube.com  https://*.ytimg.com;         frame-src 'self'
https://googleads.g.doubleclick.net https://*.youtube-nocookie.com https://*.vimeo.com
https://kiwiirc.com https://www.google.com https://fightforthefuture.github.io;
```

Figure 5: Attempts with different payload

We can see burp suite is sending requests with different payload to perform successful authentication. However we can monitor that there is not changes in the response html, which means none of the attempts are not successful yet.



Intruder attack 1
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items

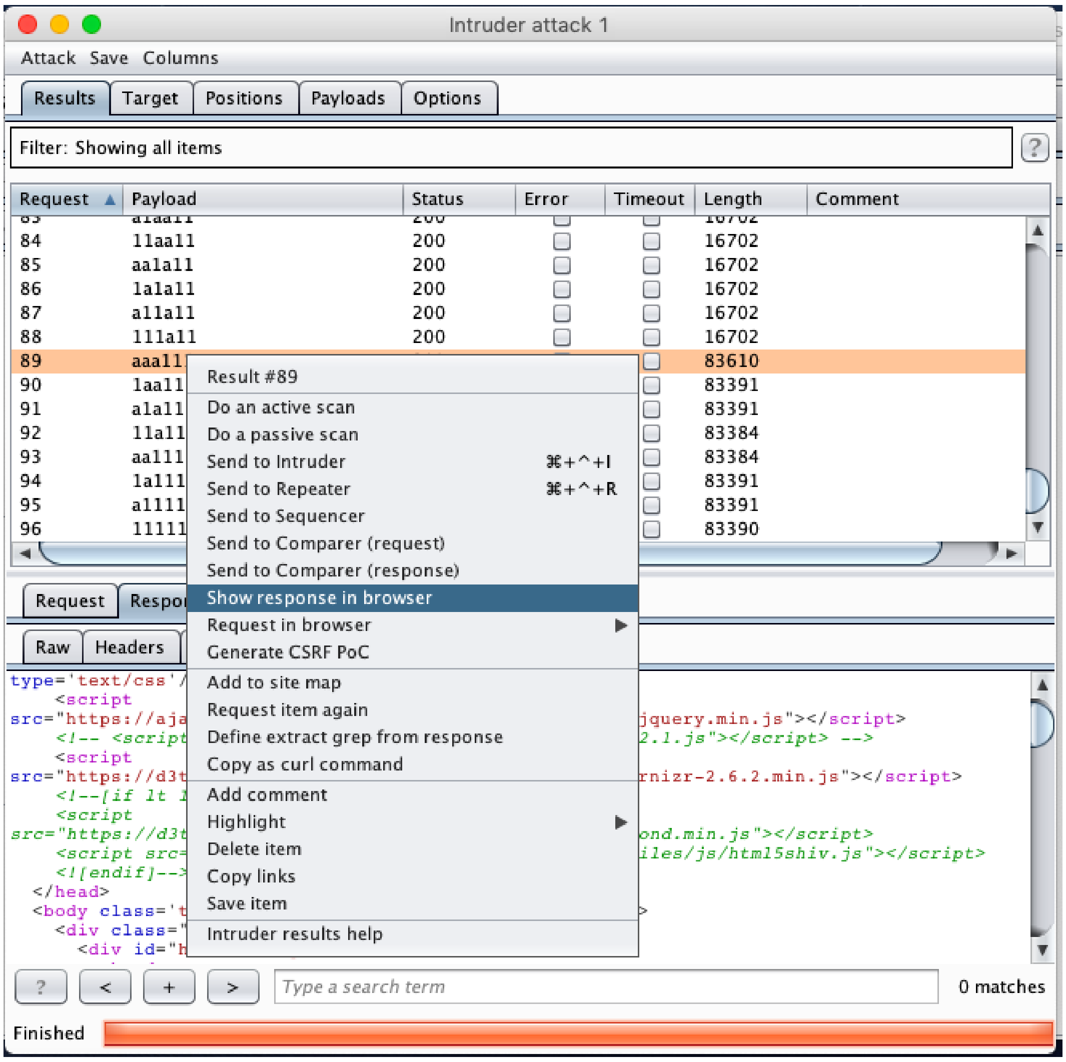
Request	Payload	Status	Error	Timeout	Length	Comment
83	a1aa11	200	☐	☐	16702	
84	11aa11	200	☐	☐	16702	
85	aa1a11	200	☐	☐	16702	
86	1a1a11	200	☐	☐	16702	
87	a11a11	200	☐	☐	16702	
88	111a11	200	☐	☐	16702	
89	aaa111	302	☐	☐	83610	
90	1aa111	200	☐	☐	83391	
91	a1a111	200	☐	☐	83391	
92	11a111	200	☐	☐	83384	
93	aa1111	200	☐	☐	83384	
94	1a1111	200	☐	☐	83391	
95	a11111	200	☐	☐	83391	
96	111111	200	☐	☐	83390	

Request Response
Raw Headers Hex HTML Render

```
type='text/css'/>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <!-- <script src="https://cdn.socket.io/socket.io-1.2.1.js"></script> -->
    <script
src="https://d3t63mlrxnixd2.cloudfront.net/files/js/modernizr-2.6.2.min.js"></script>
    <!--[if lt IE 9]>
    <script
src="https://d3t63mlrxnixd2.cloudfront.net/files/js/respond.min.js"></script>
    <script src="https://d3t63mlrxnixd2.cloudfront.net/files/js/html5shiv.js"></script>
    <![endif]-->
  </head>
  <body class='theme-dark ' data-username='' data-key=''>
    <div class="page-wrap">
      <div id="header-wrap" class="container clr">
```

Figure 6: Successful Login attempt

AS we can see, the status is 302, that means after successful login the bot was redirected to a different page. And also the length of the response is 83610 compared to previous length of 16702.

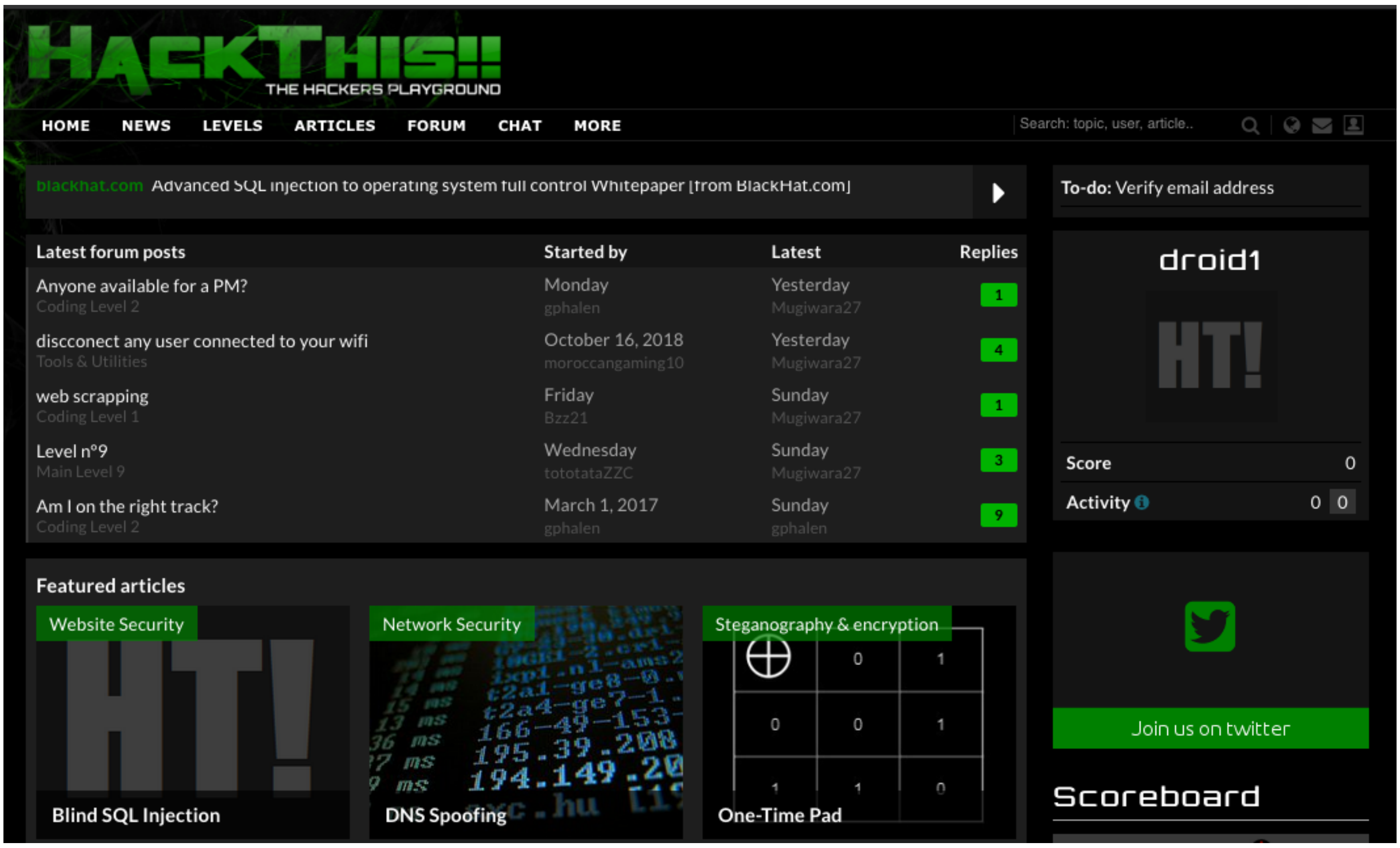


Request	Payload	Status	Error	Timeout	Length	Comment
84	1laall	200	☐	☐	16702	
85	aalall	200	☐	☐	16702	
86	1alall	200	☐	☐	16702	
87	allall	200	☐	☐	16702	
88	111all	200	☐	☐	16702	
89	aaall			☐	83610	
90	1aall			☐	83391	
91	alall			☐	83391	
92	11all			☐	83384	
93	aalll			☐	83384	
94	1alll			☐	83391	
95	allll			☐	83391	
96	11111			☐	83390	

```
type='text/css'/
    <script
src="https://aja                         jquery.min.js"></script>
    <!-- <script                         2.1.js"></script> -->
    <script
src="https://d3t                         rnizr-2.6.2.min.js"></script>
    <!--[if lt I
    <script
src="https://d3t                         ond.min.js"></script>
    <script src=                         iles/js/html5shiv.js"></script>
    <![endif]-->
  </head>
  <body class='t                         >
    <div class="
      <div id="h
```

Figure 7: Replicate in browser

We can replicate the request keeping all the arguments same. This option will recreate the request in browser, so we should be able to login the website.


HOME NEWS LEVELS ARTICLES FORUM CHAT MORE

blackhat.com Advanced SQL injection to operating system full control Whitepaper [from BlackHat.com]

To-do: Verify email address

Latest forum posts	Started by	Latest	Replies
Anyone available for a PM? Coding Level 2	Monday gphalen	Yesterday Mugiwara27	1
discconect any user connected to your wifi Tools & Utilities	October 16, 2018 moroccangaming10	Yesterday Mugiwara27	4
web scrapping Coding Level 1	Friday Bzz21	Sunday Mugiwara27	1
Level n°9 Main Level 9	Wednesday tototataZZC	Sunday Mugiwara27	3
Am I on the right track? Coding Level 2	March 1, 2017 gphalen	Sunday gphalen	9

droid1

Score 0

Activity 0 0

Featured articles

Website Security — Blind SQL Injection

Network Security — DNS Spoofing

Steganography & encryption — One-Time Pad

Join us on twitter

Scoreboard

Figure 8: Website with user details

We can see we were able to login in the website using the credential of user 'droid1'.

Burp suite is a phenomenally user friendly tool. What we have demonstrated here is just a subset of many features. Some of the features are HTTP Proxy, Scanner, Intruder, Spider, Repeater, Decoder, Comparer. We have only used Intruder functionality which can perform automated attacks on web applications. The tool offers a configurable algorithm that can generate malicious HTTP requests. It can test and detect SQL Injections, Cross Site Scripting, parameter manipulation and vulnerabilities susceptible to brute-force attacks.

2 Question

Read the article "Reflections on Trusting Trust" by K. Thompson, Communications of the ACM, Vol. 27, No. 8, pp. 761–763, 1984 (available from www.buffalo.edu/m̃blanton/cse565/rtt.pdf). Write your reaction and thoughts. Answer the following questions:

a **What is the key to a program that, when executed, reproduces itself?**

For the compiler to produce a self-replicating program or Quine it must have extensive support for string manipulation.

"This program can contain an arbitrary amount of excess baggage that will be reproduced along with the main algorithm. In the example, even the comment is reproduced." -Ken Thompson

```
char s[] = {'\t','0','\n','}',';','\n','\n','#','i','n','c','l','u','d','e',
' ','<','s','t','d','i','o','.','h','>','\n','i','n','t',' ','m','a','i','n',
'(',')',' ','{','\n','\t','i','n','t',' ','i',';','\n','\t','p','r','i','n',
't','f','(','"','c','h','a','r',' ','s','[',']',' ','=',' ','{','\\','n','"',
')',';','\n','\t','f','o','r','(','i',' ','=',' ','0',';',' ','s','[','i',
']',';',' ','i','+','+',')',' ','p','r','i','n','t','f','(','"','\\','t','%',
'd',',','\\','n','"',',',' ','s','[','i',']',')',';','\n','\t','p','r','i',
'n','t','f','(','"','%','s','"',',',' ','s',')',';','\n',' ',' ',' ',' ','r',
'e','t','u','r','n',' ','0',';','\n','}','\n',0};

```

```
#include<stdio.h>
int main() {
        int i;
        printf("char s[] = {\n");
        for(i = 0; s[i]; i++) printf("\t%d,\n", s[i]);
        printf("%s", s);
    return 0;
}
```

We can see from the above quine code that, the individual characters in the string can be escaped to represent unprintable characters like newline, tab etc. The string handling of the compiler should be comprehensive enough to handle this, using which self-replicating program can be generated.

b **How do the observations in the article relate to rootkits in operating system security?**

A rootkit is a collection of computer software, typically malicious, designed to enable access to a computer or areas of its software that is not otherwise allowed (for example, to an unauthorized user) and often masks its existence or the existence of other software. In the stage III section of the paper Ken Thompson showed us an conceptual implementation of rootkit. He programmed to Tojan horses which will give unauthorized access in restricted data, validation the user authentication. However, this approach in source is conspicuous and easily fixable.

He proposed an way around it. First we compile the modified source with the normal C compiler to produce a bugged binary. Then binary has to be installed as official C. Now the trojan can be removed from the source and the new binary will reinsert the bugs whenever it is compiled. This way the backdoor will be there without a trace.

3 Question

Consider a distributed variant of the attack we explore in Problem 7.1. Assume the attacker has compromised a number of broadband-connected residential PCs to use as zombie systems. Also assume each such system has an average uplink capacity of 128 Kbps. What is the maximum number of 500-byte ICMP echo request (ping) packets a single zombie PC can send per second? How many such zombie systems would the attacker need to flood a target organization using a 0.5-Mbps link? A 2-Mbps link? Or a 10-Mbps link? Given reports of botnets composed of many thousands of zombie systems, what can you conclude about their controller's ability to launch DDoS attacks on multiple such organizations simultaneously? Or on a major organization with multiple, much larger network links than we have considered in these problems?

Each user has uplink capacity of 128 Kbps = 128 * 1024 = 131072 bits per second

Number of maximum number of 500-byte ICMP echo request (ping) packets a single zombie PC can send per second = 131072/ (500*8) = 32.768 ≈ 32 requests.

for organization using 0.5 Mbps the attacker will need: (0.5*1024*1024)/(128*1024) = 4 zombies

for organization using 2 Mbps the attacker will need: (2*1024*1024)/(128*1024) = 16 zombies

for organization using 10 Mbps the attacker will need: (10*1024*1024)/(128*1024) = 80 zombies

Ability to mount DDoS attack on multiple such organizations simultaneously will depend on the size of the botnet and size of organizations. It is certainly possible to attack multiple organizations at the same time.

Attacking on a major organization with multiple, much larger will pose the same question again. That is the size of botnet and network size of the organization. Though it is hard to defend against DDoS attack, the large organization are expected to employ defensive mechanism like ingress filtering, SYN cookies, packet marking etc

4 Question

Based on the discussion of firewall rules in class, write firewall packet filtering rules for a corporate network with IP addresses 219.33.*.*. The firewall should permit all outgoing connections, incoming connections to the web servers with IP addresses 219.33.12.2 and 219.33.3.4 for both HTTP and HTTPS, incoming connections to the email server with IP address 219.33.12.2, and incoming SSH connections to machines with IP addresses 219.33.49.12 and 219.33.3.8.

```
# permit all outgoing connections, incoming connections
# to the web servers with IP addresses 219.33.12.2 and
# 219.33.3.4 for both HTTP and HTTPS

allow TCP *:* -> 219.33.12.2:80
allow TCP *:* -> 219.33.3.4:80
allow TCP *:* -> 219.33.12.2:443
allow TCP *:* -> 219.33.3.4:443

# incoming connections to the email server with IP address 219.33.12.2

allow TCP *:*/out -> 219.33.12.2:587/in

# incoming SSH connections to machines with IP addresses 219.33.49.12 and 219.33.3.8

allow TCP *:*/out -> 219.33.49.12:22/in
allow TCP *:*/out -> 219.33.3.8:22/in

drop * *:* -> *:*
```

References

[1] 40+ intentionally vulnerable websites to (legally) practice your hacking skills, 2016.

[2] William Stallings; Lawrie Brown. *Computer Security: Principles and Practice (3rd Edition)*. Abe Books, 2014.

[3] Wikipedia. Rootkit. https://en.wikipedia.org/wiki/Rootkit.