CS-501

Introduction to Malware, Threat Hunting
& Offensive Capabilities Development

Lecture 0x0b:
HTTP

Capstone: Developing your own C2 Framework

To pass this class, you must develop a C2 Framework.

You may work in groups of up to 5 people.

You may not work alone. Infosec is a team sport. Go make some friends :-)

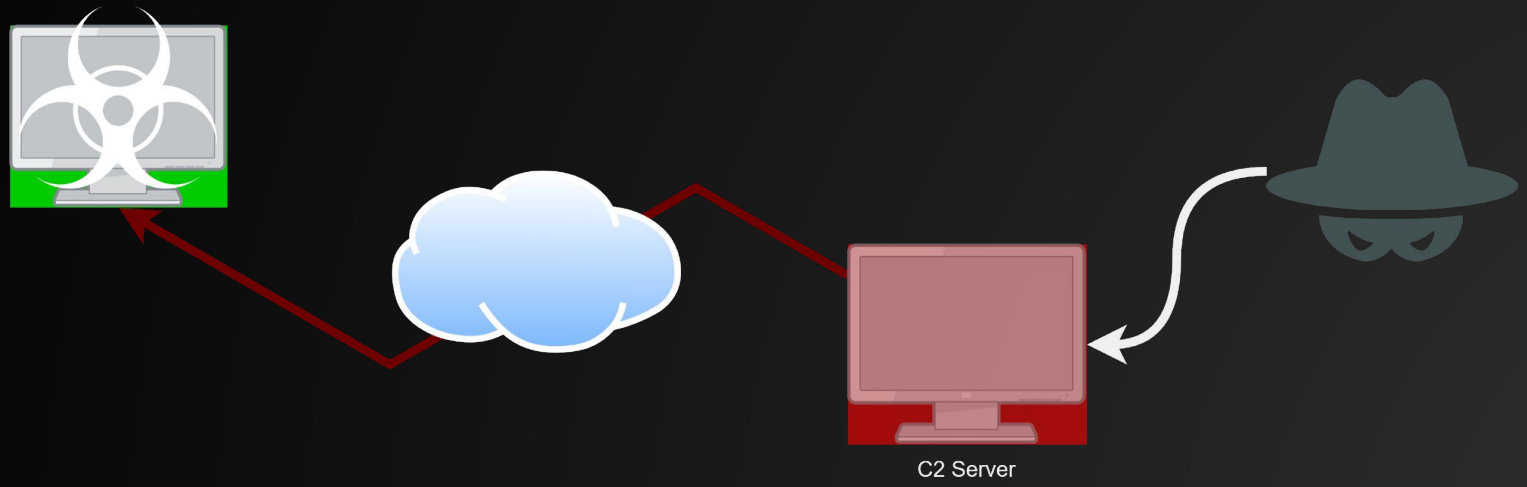
The C2 framework must support an implant that communicates using HTTP

All homework going forward is relevant to the capstone :)

Example C2 Architecture

Infected Machine connects directly to the C2

Commands are issued by the operator



Terminology: Review

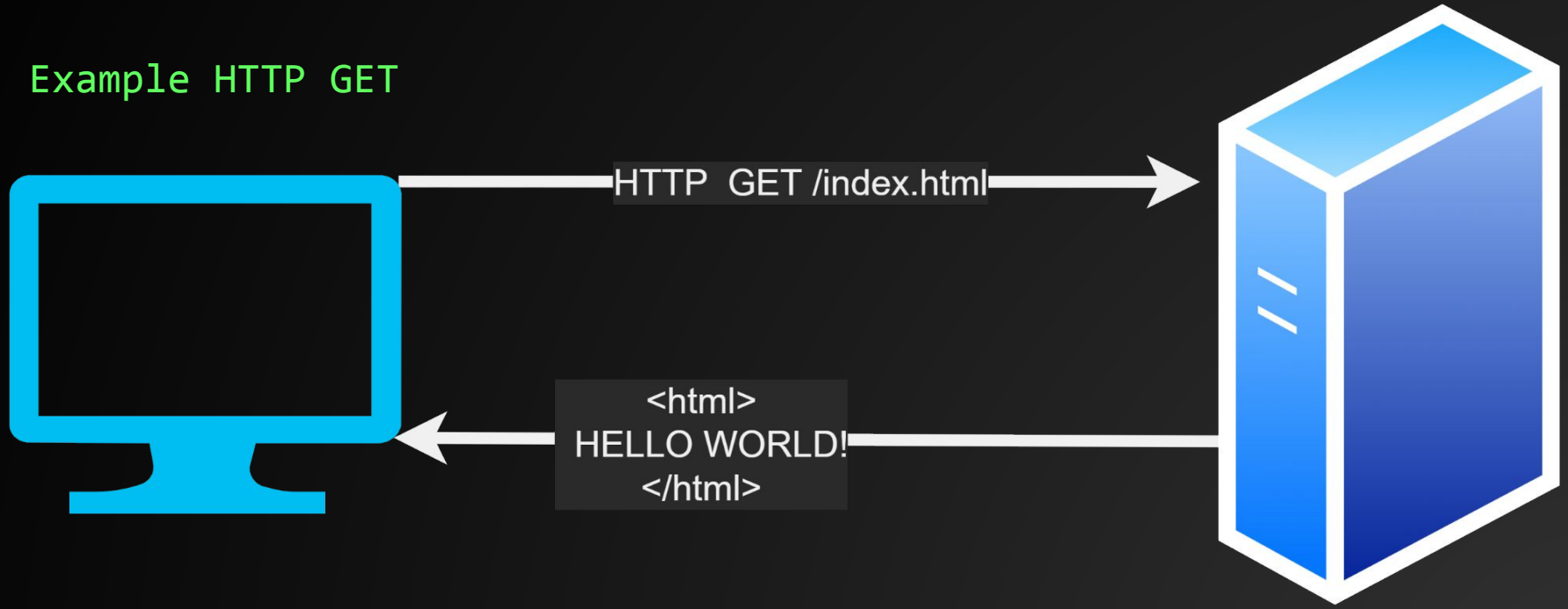
C2: Command and Control Server

Implant: the malware “implanted” on a victim machine

C2 Channel: The transport mechanism used by the implant and the C2 server to exchange data

C2 Channel: HTTP

Example HTTP GET



HTTP

- Hypertext Transfer Protocol (HTTP)
- Protocol built on top of TCP that powers the web
- Used for (among other things) transferring data between a **client and a server**

Think of TCP as a stream of data.

Think of HTTP as a sequence of envelopes of data

Has multiple different versions and extensions

HTTPs

Hypertext Transfer Protocol Secure (HTTPs) is the secure extensions of HTTP

This requires the Remote server to authenticate itself to the client using Transport layer Security (TLS) which is built on top of TCP with Public Key Infrastructure (PKI)

We will go into further detail once we get to the cryptography section, but PKI & TLS are prerequisites for this class.

Tools to Play with TCP

- netcat
- socat (way better)
  - See https://www.redhat.com/sysadmin/getting-started-socat
- Python3 socket

Socat Example

tmux

```
→ ~ socat -u TCP4-LISTEN:3334,reuseaddr,fork OPEN:/tmp/test.log,creat,append
```

```
→ ~ tail -f /tmp/test.log

hello there
hello
there
what is good
my dude
123456789
```

```
→ ~ socat - TCP4:127.0.0.1:3334
hello
there
```

```
→ ~ socat - TCP4:127.0.0.1:3334
what is good
my dude
123456789
```

[0] 0:socat* "user" 10:28 25-Oct-22

Tools to Play with HTTP (Client)

- socat
- curl
- Wget
- Fiddler2
- postman
- Python3 requests
- BurpSuite (recommended to get started)

Anatomy of a URL

<protocol>://<domain>:<port>/<path>

For example:

http://example.com/index.html

(URL is a type of a URI)

For more see
https://www.geeksforgeeks.org/difference-between-url-and-uri/

URL

HTTP → port 80

HTTPS→ port 443

These are conventions used by browsers and most HTTP clients

I.e. http://example.com → http://example.com:80/ are the same thing

Anatomy of an HTTP Request

<Verb> <URI> <HTTP Version>

<headers>...

<BODY>

HTTP Headers

Key/Value stores of data

Common headers: Cookies, host

Can be whatever you want!

BurpSuite

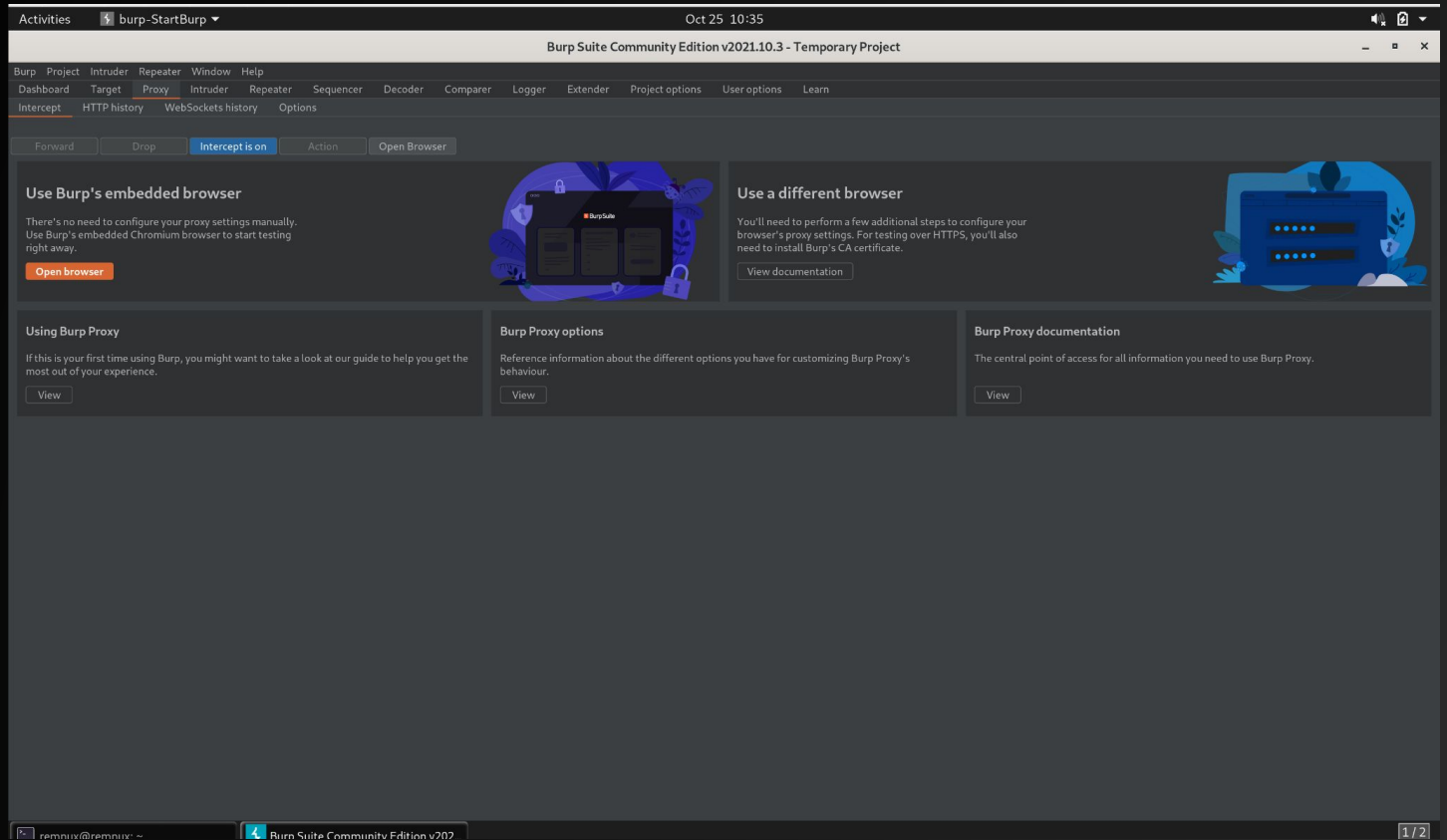
- Web penetration testing tool
- Has a free community version
- Intuitive UI, written in java
- Alternatives: Zap (big fan)
- Comes shipped with a preconfigured browser to strip TLS

Please read https://portswigger.net/burp/documentation/desktop/penetration-testing

Basics of BurpSuite

- Proxy attack framework that proxies all HTTP traffic through a MITM proxy to allow for HTTP request introspection/modification
- By using a rogue certificate, HTTPs traffic can be decrypted
- Main tabs are
  - Proxy: toggle intercept and proxy options
  - Repeater: replay HTTP requests (possibly modified)

BurpSuite


Activities
burp-StartBurp
Oct 25 10:35
Burp Suite Community Edition v2021.10.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Use Burp's embedded browser
There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.
Open browser
Use a different browser
You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.
View documentation
Using Burp Proxy
If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.
View
Burp Proxy options
Reference information about the different options you have for customizing Burp Proxy's behaviour.
View
Burp Proxy documentation
The central point of access for all information you need to use Burp Proxy.
View
remnux@remnux: ~
Burp Suite Community Edition v202
1 / 2

HTTP Verbs

Verb is the action the client wants the server to perform

These are conventions!

Basic Verbs

GET: fetch content from the server. Typically the client request has no HTTP body and the server returns content in the HTTP body

POST: Submit data to the server, often times creating side effects. Data is submitted in the client HTTP body

For more see https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

HTTP GET

- Retrieve content from a remote server
- This is what happens whenever you type a URL into web browser like chrome

Example

```
socat - TCP4:example.com:80
GET / HTTP/1.1
Host: example.com
```

```
➜  Downloads socat - TCP4:example.com:80
GET / HTTP/1.1
Host: example.com

HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 475417
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Tue, 25 Oct 2022 15:04:56 GMT
Etag: "3147526947"
Expires: Tue, 01 Nov 2022 15:04:56 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (bsa/EB16)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256

<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
        color: #38488f;
        text-decoration: none;
    }
    @media (max-width: 700px) {
        div {
            margin: 0 auto;
            width: auto;
        }
    }
    </style>
</head>
```

Activities burp StartBurp Oct 25 10:54

Burp Suite Community Edition v2021.10.3 - Temporary Project

Burp Project Intruder Repeater Window Help

Sequencer Decoder Comparer Logger Extender Project options User options Learn

Dashboard Target Proxy Intruder Repeater

Intercept HTTP history WebSockets history Options

Request to https://example.com:443 [93.184.216.34]

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
GET / HTTP/1.1
Host: example.com
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="96"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

INSPECTOR

Search... 0 matches

Burp Suite

example.com

Burp Suite

Keep up with the latest vulnerabilities

Web Security Academy

Register for free to advance your skills with interactive challenges from our leading researchers.

Get started →

Familiarize yourself with Burp Suite

Learn about Burp Suite and its main tools with our videos, guides and documentation.

Video tutorials →

Burp documentation →

Upgrade to Burp Suite Professional

Unlock your potential.

Access the industry trusted Burp Scanner, unthrottled Burp Intruder, and more.

Upgrade now →

Activities burp-StartBurp Oct 25 10:57

Burp Suite Community Edition v2021.10.3 - Temporary Project

Burp Project Intruder Repeater Window Help

Sequencer Decoder Comparer Logger Extender Project options User options Learn

Dashboard Target Proxy Intruder Repeater

Intercept HTTP history WebSockets history Options

Forward Drop Intercept is on Action Open Browser

Example Domain

Not secure | example.com

Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.

More information...

POST

Send data to the server. Often times used in HTML forms.

I.e., when you log into a website

Activities burp-StartBurp Oct 25 11:16

Burp Suite Community Edition v2021.10.3 - Temporary Project

Burp Project Intruder Repeater Window Help

Sequencer Decoder Comparer Logger Extender Project options User options Learn

Dashboard Target Proxy Intruder Repeater

Intercept HTTP history WebSockets history Options

Request to https://shib.bu.edu:443 [54.204.187.34]

Forward Drop Intercept is on Action Open Browser Comment this item HTTP/2

Pretty Raw Hex

```
POST /idp/profile/SAML2/Redirect/SSO?execution=e1s1 HTTP/2
Host: shib.bu.edu
Cookie: JSESSIONID=hyo6a0bsbu1kxdw4zi8l4ep1; AWSALB=
5IgTIBGcMMdlKTdbpaVo4ufKHFMFTkhlAADm7CaEjjP3+DJgLoGtki/uACiIbLbD5Ty2kpmrPvp3ppnZjL21K47SIYZlhUu5yqHDLgZdtHwxPE
0XnmjvMnV0BG2J; AWSALBCORS=
5IgTIBGcMMdlKTdbpaVo4ufKHFMFTkhlAADm7CaEjjP3+DJgLoGtki/uACiIbLbD5Ty2kpmrPvp3ppnZjL21K47SIYZlhUu5yqHDLgZdtHwxPE
0XnmjvMnV0BG2J
Content-Length: 74
Cache-Control: max-age=0
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="96"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
Origin: https://shib.bu.edu
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/96.0.4664.45 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/s
igned-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://shib.bu.edu/idp/profile/SAML2/Redirect/SSO;jsessionid=hyo6a0bsbu1kxdw4zi8l4ep1?execution=e1s1
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

j_username=ronaldfillabuster&j_password=iamverysecure%21&_eventId_proceed=
```

Search... 0 matches

INSPECTOR

Boston University | Login

https://shib.bu.edu/idp/profile/SAML2/Redirect/SSO;jsessionid=hyo6a0bsbu1kxdw4zi8l4ep1?execution=e1s1#

BU Login

ronaldfillabuster

••••••••••••••

Logging in, please wait...

You have asked to login to google.com/a/bu.edu

Forgot Password

Update Your Account

Login Help

Anatomy of a Server Response

```
<HTTP Version> <Status Code> <Message>

<HTTP headers>...

<BODY>
```

Server Side Status Codes

Integer corresponding to the status of an HTTP Request

Returned by the server to tell the client if everything is good

2xx OK

3xx Go somewhere else

4xx You f*cked up

5xx I f*cked up

We will not see a handful of status codes you will likely encounter during the project.

2xx

200: I (the server) have successfully serviced your request

3xx

Usually used as part of a redirect to another resource

You might see this if you try to go to /profile in the CTF and you are not signed in

The server then redirects you to the /login resource

4xx

You (the client) have made a mistake and I won’t service you

400 Bad Request: the client sent data to the server in a format it doesn’t understand

401: You are not logged in and I won’t service you

403 Forbidden: You are authenticated but I still won’t service you

404 Not Found: You asked for a resource that doesn’t exist

405 Method not allowed: the client used an unsupported HTTP Verb

5xx

500 Internal Server Error: Catch all error for the server encountered an error while trying to service your request

502 Bad Gateway: You are using a reverse proxy that tried to proxy a request to a server that it could not connect to

FOr more see https://bobcares.com/blog/502-bad-gateway-nginx/

For more information…

See https://developer.mozilla.org/en-US/docs/Web/HTTP/Status for more information about status codes.

Activities burp-StartBurp Oct 25 10:58

Burp Suite Community Edition v2021.10.3 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

2 ...

Send Cancel

Target: https://example.com HTTP/2

Request

Pretty Raw Hex

```
1 GET / HTTP/2
2 Host: example.com
3 Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="96"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/96.0.4664.45 Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
  .8,application/signed-exchange;v=b3;q=0.9
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate
14 Accept-Language: en-US,en;q=0.9
15 Connection: close
16
17
```

Search... 0 matches

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Accept-Ranges: bytes
3 Age: 309651
4 Cache-Control: max-age=604800
5 Content-Type: text/html; charset=UTF-8
6 Date: Tue, 25 Oct 2022 14:58:09 GMT
7 Etag: "3147526947"
8 Expires: Tue, 01 Nov 2022 14:58:09 GMT
9 Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
10 Server: ECS (bsa/EB24)
11 Vary: Accept-Encoding
12 X-Cache: HIT
13 Content-Length: 1256
14
15 <!doctype html>
16 <html>
17   <head>
18     <title>
       Example Domain
     </title>
19
20     <meta charset="utf-8" />
21     <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
22     <meta name="viewport" content="width=device-width, initial-scale=1" />
23     <style type="text/css">
24       body{
25         background-color:#f0f0f2;
26         margin:0;
27         padding:0;
28         font-family:-apple-system,system-ui,BlinkMacSystemFont,"Segoe UI","Open Sans",
           "Helvetica Neue",Helvetica,Arial,sans-serif;
29
30       }
31       div{
32         width:600px;
33         margin:5emauto;
34         padding:2em;
35         background-color:#fdfdff;
36         border-radius:0.5em;
37         box-shadow:2px3px7px2pxrgba(0,0,0,0.02);
38       }
39       a:link,a:visited{
40         color:#38488f;
41         text-decoration:none;
42       }
43       @media(max-width:700px){
44         div{
45           margin:0auto;
46           width:auto;
47         }
48       }
```

Search... 0 matches

INSPECTOR

- Request Attributes
- Query Parameters (0)
- Body Parameters (0)
- Request Cookies (0)
- Request Headers (17)
- Response Headers (12)

Done 1,605 bytes | 12 millis

35

HTTP Server

Server to service HTTP requests

This course will use Flask to implement the HTTP server

It is relatively bare bones, but has a rich ecosystem

Core component is a Flask application that you create routes and handlers for

The handler will be invoked whenever a request is made to the specified route

Flask

- Lightweight, no frills HTTP server
- Routing is handled by decorators
- Contains various helper functions to easily parse and respond to requests
- Has a rich ecosystem for different database plugins
- Is not production ready in and of itself-- requires a Web Service Gateway Interface (WSGI)

Example: Hello world

```
from flask import Flask, request

app = Flask(__name__)

@app.route("/hello")
def hello():
    return "Hello world!"
```

Learning Flask

You are expected to read through a flask tutorial

For example: https://www.tutorialspoint.com/flask/index.htm

Often times, googling “how do i do x in Flask” will yield you a usable answer

HTTP RPC

RPC: Remote Procedure Call

RPC is the protocol used to control a remote machine.

The remote machine exposes some interface that contains a collection of functions that can be remotely invoked.

Example: Double it

Create a web server that handles the endpoint /double/<some int>

And returns the doubled integer along with a helpful message

This is an example of an RPC!

RPC in malware

With malware, the RPC is exposed by implant! This the opposite of what we just did where the HTTP server executes the commands!

Let’s consider the simple example of malware that only wishes to execute a handful of commands on a victim machine

Example: Whoami

Consider an RPC that exposes the following function:

whoami: get the current user on the computer and send it back

This can be implemented in HTTP as follows:

Server: Expose /whoami (GET) to task the client to execute whoami

Server: Expose /whoami (POST) to receive the results

Client: CHeck for tasks using GET /whoami. If the task is whoami: execute whoami and send the results in HTTP post

Components of an RPC

Transport: The Protocol used to send and receive data

Message Format: the way in which data is structured

Exported Interfaces/Functions: the suite of functions supported by the RPC framework

RPC Example: Restful APIs

- Transport is HTTP(s)
- Format is JavaScript Object Notation (JSON)

RESTFUL Example

RPC:

whoami: Args→ None. Gets the current user

hostname: Args→None. Gets the hostname

double: Args→List of integers to double

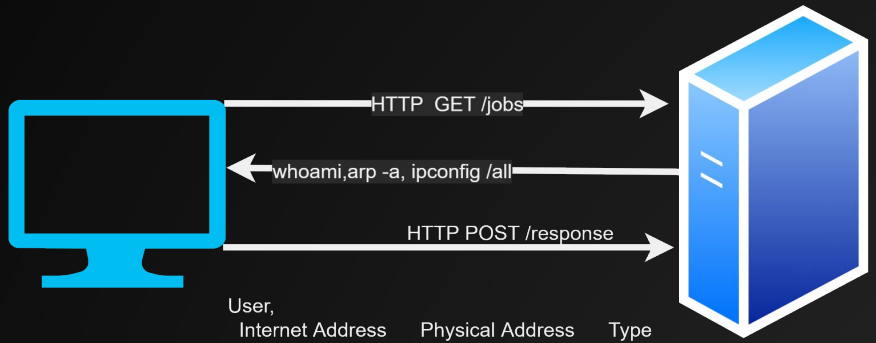
Example 1: HTTP Reverse Shell

Malware makes an HTTP GET Request to the endpoint /commands

Server responds with a list of shell commands it wishes the implant to execute

Malware responds with a post request containing the output of those commands

Example 1: HTTP Reverse Shell



User,

Internet Address	Physical Address	Type
10.16.0.1		dynamic
10.16.255.255		static
224.0.0.22		static
224.0.0.251		static
224.0.0.252		static
239.255.255.250		static
255.255.255.255		static

```
Connection-specific DNS Suffix  . :
Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter #2
Physical Address. . . . . . . . . : 08-00-27-D6-2A-9E
DHCP Enabled. . . . . . . . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::a47d:1f11:8f29:22e6%9(Preferred)
IPv4 Address. . . . . . . . . . . : 10.10.10.3(Preferred)
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 10.10.10.2
DHCPv6 IAID . . . . . . . . . . . : 319291431
DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-28-58-BD-84-00-15-5D-24-F4-CD
DNS Servers . . . . . . . . . . . : 10.10.10.2
NetBIOS over Tcpip. . . . . . . . : Enabled
```

C2 Engineering Basics

Send data

Get data

Profit

Team Server

Consider the following:

Multiple implants connect to the C2

Multiple operators connect to the C2 to control the agents

Operators need to be appraised of the activity of their colleagues and status updates of their agents

Messages are sent from the teamserver to operators

- Example messages that operators would want to see
- A new bot has connected to the server
- A bot has pulled down a task
- A bot has sent data to the server
- An operator has issued a command to an agent

We will work up to this

Server Components and Concepts

Listeners: a server that handles connections from implants

HTTP Listener: a listener that services implants using HTTP endpoints

Components of a C2

3 programs:

Implant: the malicious code that implements the RPC

Teamserver: the server controlling the implant

Client: the client code used by the operator to communicate with the Teamserver. Thus, allowing the client to control the implant.

Discussion:
Drawing out a C2 framework

Basic Imports for our teamserver

```
from flask import Flask, request, jsonify
```

Flask: flask application

request: http request object

jsonify: method to create a json response

HTTP Client Windows C++

WinSock → TCP client but you can implement HTTP

WinHTTP → Commonly used by malware. **Easy to make proxy aware**

WinInet → less common but still popular (i.e. Cobalt Strike) **easy to make proxy aware with authentication**

External Library: Mixed results. Libcurl is a great choice but it requires statically linking a TLS library

Advanced Concepts for your C2

- Implant identification
- Job Identification
- Flask Blueprints
- Messaging
- Databases
- Operator authentication